Department of IT & Computer Application
M.M.M. University of Technology
Gorakhpur

Credit structure for Two-Year MCA Programme
(For newly admitted students from session 2020-21)

Category	I	II	III	IV	Total
Programme Core (PC)	20	22	22	5	**69**
Programme Electives (PE)	-	-	4	4	**8**
Project (P)	-	-	-	10	**10**
Audit	-	-	-	-	-
Total	**20**	**22**	**26**	**19**	**87**

Junior Year, Semester-I

Sr.	Category	Paper Code	Subject Name	L	T	P	Cr.
1	PC	MCA-111	Object Oriented Programming with C^{++}	3	1	2	**5**
2	PC	MCA-112	Database Management Systems	3	1	2	**5**
3	PC	MCA-113	Computer Organization & Architecture	3	1	-	**4**
4	PC	MCA-114	Software Engineering	3	1	-	**4**
5	PC	MCA-115	Software Lab-1	-	1	2	**2**
6	AC	Audit Subject	-	-	-	-	-
Total				**12**	**5**	**6**	**20**

Junior Year, Semester-II

Sr.	Category	Paper Code	Subject Name	L	T	P	Cr.
1	PC	MCA-201	JAVA Programming	3	1	2	**5**
2	PC	MCA-202	Data Structures & Applications	3	1	2	**5**
3	PC	MCA-203	Web Technologies	3	1	2	**5**
4	PC	MCA-204	Operating System Concepts	3	1	2	**5**
5	PC	MCA-205	Software Lab-2	-	1	2	**2**
6	AC	Audit Subject	-	-	-	-	-
Total				**12**	**5**	**10**	**22**

Senior Year, Semester-III

Sr.	Category	Paper Code	Subject Name	L	T	P	Cr.
1	PC	MCA-301	Artificial Intelligence	3	1	2	**5**
2	PC	MCA-302	Computer Network	3	1	2	**5**
3	PC	MCA-303	Design & Analysis of Algorithms	3	1	2	**5**
4	PC	MCA-304	Cloud Computing	3	1	2	**5**
5	PE1	MCA-***	Elective-I	3	1/0	0/2	**4**
6	PC	MCA-305	Mini Project	-	-	4	**2**
7	AC	MCA-Ind	Industrial/Practical Training	-	-	2	-
Total				**15**	**5/4**	**12/14**	**26**

Senior Year, Semester-IV

Sr.	Category	Paper Code	Subject Name	L	T	P	Cr.
1	PC	MCA-401	Internet of Things	3	1	2	**5**
2	PE2	MCA-***	Elective-II	3	1/0	0/2	**4**
3	AC	MCA-Sem	Seminar	-	-	6	-
4	P	MCA-Prj	Project	-	-	20	**10**
Total				**6**	**2/1**	**22/24**	**19**

Programme Core (PC) for MCA

Sr.	Category	Paper Code	Subject Name	L	T	P	Cr.
1.	PC	MCA-111	Object Oriented Programming with c++	3	1	2	**5**
2.	PC	MCA-112	Database Management Systems	3	1	2	**5**
3.	PC	MCA-113	Computer Organization & Architecture	3	1	-	**4**
4.	PC	MCA-114	Software Engineering	3	1	-	**4**
5.	PC	MCA-115	Software Lab-1	-	1	2	**2**

6.	PC	MCA-201	JAVA Programming	3	1	2	**5**
7.	PC	MCA-202	Data Structures & Applications	3	1	2	**5**
8.	PC	MCA-203	Computer Network	3	1	2	**5**
9.	PC	MCA-204	Operating System Concepts	3	1	2	**5**
10.	PC	MCA-205	Software Lab-2	-	1	2	**2**
11.	PC	MCA-301	Artificial Intelligence	3	1	2	**5**
12.	PC	MCA-302	Web Technologies	3	1	2	**5**
13.	PC	MCA-303	Design & Analysis of Algorithms	3	1	2	**5**
14.	PC	MCA-304	Cloud Computing	3	1	2	**5**
15.	PC	MCA-305	Mini Project	-	-	4	**2**
16.	PC	MCA-401	Internet of Things	3	1	2	**5**
Total				**39**	**15**	**30**	**69**

Programme Electives (PE-I)

Sr.	Paper Code	Subject Name	L	T	P	Cr.
1	MCA-351	Automata Theory	3	1	-	**4**
2	MCA-352	Python Programming	2	1	2	**4**
3	MCA-353	Information Security & Cyber Laws	3	1	-	**4**
4	MCA-354	Android Programming	2	1	2	**4**
5	MCA-355	Distributed Database Systems	3	1	-	**4**
6	MCA-356	Computer Graphics	3	1	-	**4**
7	MCA-357	Distribution Systems	3	1	-	**4**
8	MCA-358	Data Mining & Warehousing	3	1	-	**4**

Programme Electives (PE-II)

Sr.	Paper Code	Subject Name	L	T	P	Cr.
1	MCA-451	Compiler Design	2	1	2	**4**
2	MCA-452	Machine Learning	2	1	2	**4**
3	MCA-453	Wireless Sensor Networks	3	1	-	**4**
4	MCA-454	Mobile Computing	3	1	-	**4**
5	MCA-455	Big Data Technologies	3	1	-	**4**
6	MCA-456	Data Sciences & Analysis	3	1	-	**4**
7	MCA-457	Natural Language Processing	2	1	2	**4**
8	MCA-458	Blockchain & Cryptocurrency	3	1	-	**4**

Audit Courses for MCA

Sr.	Paper Code	Subject Name	L	T	P	Cr.
1	BAS-03	Professional Communication	3	1	-	**4**
2	BAS-24	Applied Computational Methods	3	1	2	**5**
3	MAS-103	Communication for Business Management	3	1	2	**5**
4	MAS-105	Applied Probability & Statistics	3	1	0	**4**
5	MAS-106	Discrete Mathematics	3	1	0	**4**

MCA-111 OBJECT ORIENTED PROGRAMMING WITH C++5 Credits (3-1-2)

Course Objectives:

1. Understand object-oriented programming and advanced C++ concepts.
2. Be able to program using more advanced C++ features such as composition of objects, operator overloads, dynamic memory allocation, inheritance and polymorphism, file I/O, exception handling, etc.
3. Be able to build C++ classes using appropriate encapsulation and design principles.

Learning Outcomes:

1. To understand the concept of Object-Oriented Programming and master OOP using C++.
2. Ultimate goal: to make you a good programmer
3. Improve your problem-solving skills.

UNIT I **9**

Object Oriented Programming Paradigm, Basic Concepts of Object Oriented Programming, Benefits of OOP, Object Oriented Languages, Tokens, Keywords, Identifiers and Constants, Basic Data Types, User-Defined and Derived Data Types, Type Compatibility, Reference, Variables, Scope Resolution Operator, Type Casting, Implicit Conversion, Operator Precedence, Control Structures, Structure, Function.

UNIT II **9**

Class specification, class objects, accessing class members, data hiding, empty classes, pointers within a class, passing objects as arguments, returning objects from functions, friend functions and friend classes, constant parameters and member functions, structures and classes, static members, objects and memory resource, class design steps. Constructors, destructor, constructor overloading, order of construction and destruction, constructors with default arguments, nameless objects, dynamic initialization through constructors, constructors with dynamic operations, constant objects and constructor, static data members with constructors and destructors, nested classes.

UNIT III **9**

Defining Operator Overloading, Overloading Unary Operators, Overloading Binary Operators, Overloading Binary Operators Using Friends, Manipulation of Strings Using Operators, Rules for Overloading Operators, Type Conversions, Deriving Derived Classes, Single, Multilevel, Multiple, Hierarchical, Hybrid Inheritance, constructors destructors in derived classes, constructor's invocation and data members initialization, Virtual Base Classes, Abstract Classes.

UNIT IV **9**

Pointers to Objects, Classes for File Stream Operations, Opening and Closing a File, File Pointers and their Manipulations, Sequential Input and Output Operations. Class Templates with multiple parameters, Function Templates, Overloading of Template Functions, Member Function Templates.

COMPUTER PROGRAMMING LAB

Write C++ Programs to illustrate the concept of the following:

1. Arrays
2. Structures
3. Pointers
4. Objects and Classes
5. Console I/O operations
6. Scope Resolution and Memory Management Operators
7. Inheritance
8. Polymorphism
9. Virtual Functions
10. Friend Functions
11. Operator Overloading
12. Function Overloading
13. Constructors and Destructors
14. this Pointer
15. File I/O Operations
16. Assignments on developing interfaces- multiple inheritance, extending interfaces
17. Assignments on multithreaded programming, handling errors and exceptions, applet programming and graphics programming

Books & References:

1. B. Trivedi, "Programming with ANSI C++", Oxford University Press, 2007.
2. Ira Pohl, "Object Oriented Programming using C++", Pearson Education, Second Edition Reprint
3. B. Stroustrup, "The C++ Programming language", Third edition, Pearson Education, 2004.
4. Robert Lafore, "Object Oriented Programming in Turbo C++", Galgotia Publication 1994.
5. E. Balagurusamy,"Object-Oriented Programming with C++", TMH Publication.
6. Booch, Maksimchuk, Engle, Young, Conallen and Houstan, "Object Oriented Analysis and Design with Applications", Pearson Education.
7. S. B. Lippman, Josee Lajoie, Barbara E. Moo, "C++ Primer", Fourth Edition, Pearson Education 2005.
8. Timthy Budd, "An Introduction to Object Oriented Programming with C++," Addition-Wesley.
9. Kip R. Irvine," C++ and Object-Oriented Programming," Prentice Hall

MCA-112 DATABASE MANAGEMENT SYSTEMS 5 Credits (3-1-2)

Course Objectives: Upon successful completion of the course, students

1. know and understand the fundamental concepts, techniques, and terminology of the database management system.
2. know and understand the basic syntax, semantics, and pragmatics of SQL & PL/SQL.
3. can analyze problems and apply DBMS concepts and techniques to develop appropriate programs to solve the problems,
4. can evaluate alternative database designs to determine which are better according to selected criteria,
5. know and understand the basic features of database transactions and concurrency control.
6. are able to reason about and manipulate concurrency control techniques.

Learning Outcomes:

1. list and define the fundamental concepts of database management system.
2. manually execute a given (simple) database design a transaction over it.
3. manually infer the type of a given (simple) database transaction.
4. implement (simple) algorithms and data structures as database transaction.
5. design (large) databases that are modular and have reusable components.
6. explain on a simple problem how to apply concurrency control over concurrent database transactions.

UNIT- I **9**

Introduction: An overview of database management system, database system Vs file system, Database systemconcepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure.
Data Modeling using the Entity Relationship Model: ER model concepts, notation for ER diagram, mappingconstraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree

UNIT- II **9**

Relational data Model and Language: Relational data model concepts, integrity constraints: entity integrity,referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus, **Introduction to SQL:** Characteristics of SQL. Advantage of SQL. SQL data types and literals. Types of SQL commands. SQL operators and their procedure. Tables, views and indexes. Queries and sub queries. Aggregate functions. Insert, update and delete operations. Joins, Unions, Intersection, Minus, Cursors in SQL. induction over

UNIT- III **9**

Data Base Design & Normalization: Functional dependencies, normal forms, first, second, third normal forms,BCNF, inclusion dependences, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

UNIT- IV **9**

Transaction Processing Concepts: Transaction system, Testing of serializability, Serializability of schedules,conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling. **Concurrency Control Techniques:** Concurrency control, locking techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi version schemes, Recovery with concurrent transaction.

COMPUTER PROGRAMMING LAB

1. Exercises to be based on Sybase / Oracle / Postgres / VB / Power Builder / DB2 / MS-Access.

2. Applications involving vendor development systems, stores management system, finance management etc.
3. Creation and querying of database tables for following cases. .

    a. Write SQL queries using logical operations (=,<,>,etc)
    b. Write SQL queries using SQL operators
    c. Write SQL query using character, number, date and group functions
    d. Write SQL queries for relational algebra
    e. Write SQL queries for extracting data from more than one table
    f. Write SQL queries for sub queries, nested queries
    g. Write programme by the use of PL/SQL
    h. Concepts for ROLL BACK, COMMIT & CHECK POINTS
    i. Create VIEWS, CURSORS and TRGGERS & write ASSERTIONS.
    j. Create FORMS and REPORTS
    k.
4. Design of tables by normalization and dependency analysis.
5. Writing application software with host language interface

Books & References:

1. Date C J, "An Introduction To Database System", Addision Wesley.
2. Korth, Silbertz, Sudarshan, "Database Concepts", McGraw Hill.
3. Elmasri, Navathe, "Fundamentals Of Database Systems", Addision Wesley.
4. Leon & Leon, "Database Management System", Vikas Publishing House.
5. Bipin C. Desai, "An introduction to Database Systems", Galgotia Publication.
6. Majumdar & Bhattacharya, "Database Management System", TMH.
7. Ramakrishnan, Gehrke, "Database Management System", McGraw Hill.
8. Kroenke, "Database Processing: Fundamentals, Design and Implementation", Pearson Education.
9. Maheshwari Jain, "DBMS: Complete Practical Approach", Firewall Media, New Delhi.

MCA-113 COMPUTER ORGANISATION & ARCHITECTURE 4 Credits (3-1-0)

Course Objectives:

1. To impart the essential knowledge on the fundamentals and applications of digital circuits and digital computing principles
2. To provide an overview on the design principles of digital computing systems
3. To provide technical knowledge about various digital hardware components
4. To understand how computers are constructed out of a set of functional units and how the functional units operate, interact, and communicate

Learning Outcomes:

1. Ability to understand the merits and pitfalls in computer performance measurements
2. Ability to understand memory hierarchy and its impact on computer cost/ performance

UNIT- I **9**

Data Representation, Binary, Octal, HEX and their inter-conversion, 1's and 2's complement, Binary Arithmetic, Number Systems – BCD, EBCDIC, ASCII.Basic Gates & its Truth tables, Boolean algebra, Fundamental concepts of Boolean algebra, Basic Theorem and properties of Boolean algebra, Boolean functions, Canonical and standard forms, Sum Of Product, Product of Sum, K-map method (up to 4 variables), don't care conditions, Combination circuit design with AND, OR, NOT, NAND, NOR gates, Exclusive-OR and Equivalence Functions, Universal gates functionality

UNIT- II **9**

Combinational Circuits: Half Adder, Full Adder, Binary Adder and Subtracter, Decoder / Encoder, Multiplexer /Demultiplexer.
Sequential Circuits: Flip Flops - SR, D, JK, Master – Slave, Edge Triggered, Shift Registers, Synchronous Counterand Asynchronous Counter.

UNIT- III **9**

Register Transfer Language, Bus and Memory Transfers, Bus Architecture, Bus Arbitration, Arithmetic Logic, Shift Microoperation, Arithmetic Logic Shift Unit, Design of Fast address, IEEE standard for Floating point numbers. Control Design: Hardwired & Micro Programmed Control Unit. Processor Design: Processor

Organization: General register organization, Stack organization, Addressing mode, Instruction format, Data transfer & manipulations, Program Control, Reduced Instruction Set Computer.

UNIT- IV **9**

Input-Output Organization: I/O Interface, Modes of transfer, Interrupts & Interrupt handling, Direct Memory access, Input-Output Processor, Serial Communication.
Memory Organization: Memory Hierarchy, Main Memory (RAM and ROM Chips), Auxiliary memory, Cache memory, Virtual Memory.

Books & References:

1. Digital Design by M Morris Mano, M D Ciletti
2. Computer System Architecture, by M. Mano
3. Computer Organization, by Vravice, Zaky&Hamacher

MCA-114 **SOFTWARE ENGINEERING** **4 Credits (3-1-0)**

Course Outcomes

The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Enhance the Software Project Management skills.
2. Develop functioning software which benchmarks to the international standards.

UNIT- I

Evolution and impact of Software engineering, software life cycle models: Waterfall, prototyping, Evolutionary and Spiral models. Feasibility study, Functional and Non-functional requirements, Requirements gathering, Requirements analysis and specification.

UNIT- II

Basic issues in software design, modularity, cohesion, coupling and layering, function-oriented software design: DFD and Structure chart, object modeling using UML, Object-oriented software development, user interface design. Coding standards and Code review techniques

UNIT- III

Fundamentals of testing, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, White box and black box testing, Alpha and Beta Testing of Products, Test coverage analysis and test case design techniques, mutation testing, Static and dynamic analysis, Software reliability metrics.

UNIT- IV

Software Maintenance and Software Project Management, Software as an evolutionary entity, need for maintenance, categories of maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re-Engineering, Reverse Engineering. Software Configuration Management Activities, Change Control Process, Software Version Control, An Overview of CASE Tools. Estimation of Various Parameters such as Cost, Efforts, Schedule/Duration, Constructive Cost Models (COCOMO), Resource Allocation Models, Software Risk Analysis and Management.

Textbooks

1. R. S. Pressman, "Software Engineering - A practitioner's approach", III Edition, McGraw Hill International editions, 1992.

Reference Books

1. IAN Sommerville, "Software Engineering", Pearson Education Asia, VI Edition, 2000.
2. Pankaj Jalote, "An Integrated Approach to software Engineering", Springer Verlag, 1997.

MCA-115 **Software Lab-1**

Course category : Programme Core (PC)

Pre-requisite Subject : NIL

Contact hours/week : Lecture: 0, Tutorial: 1 , Practical: 2

Number of Credits :2

Course Assessment methods : Continuous assessment through Viva-voce, Practical work/Record, attendance and Major Practical Examination

Course Outcomes : The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Understanding of Booting Process and installation of Operating system
2. Usage of Operating system commands
3. Understanding of Shell and its usage as a programming language
4. Understanding of Computer Networking concepts

Experiments

1. Understanding CMOS settings of operating system
2. Installation of Linux operating system using virtualization technique
3. Understanding and practice of various Linux commands
4. Creation/usage of various types of files supported by Linux
5. Practice of Computer networking commands
6. Programs using shell programming

MCA-201 JAVA PROGRAMMING 5 Credits (3-1-2)

Course Outcomes

The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Analyze and explain the behavior of programs involving the fundamental program constructs
2. Identify and correct syntax and logic errors in short programs
3. Design and implement a class based on attributes and behaviors of objects
4. Describe the parameter passing mechanisms in terms of formal parameters, actual parameters, non-object parameters and object parameters

UNIT- I

Object oriented programming concepts – objects – classes – methods and messages – abstraction and encapsulation – inheritance – abstract classes – polymorphism. - Objects and classes in Java – defining classes – methods - access specifiers – static members – constructors – finalize method

Arrays – Strings - Packages – Java-Doc comments –- Inheritance – class hierarchy – polymorphism – dynamic binding – final keyword – abstract classes

UNIT- II

The Object class – Reflection – interfaces – object cloning – inner classes – proxies - I/O Streams - Graphics programming – Frame – Components – working with 2D shapes.

UNIT- III

Basics of event handling – event handlers – adapter classes – actions – mouse events – AWT event hierarchy – introduction to Swing – Model-View-Controller design pattern – buttons – layout management – Swing Components – exception handling – exception hierarchy – throwing and catching exceptions.

UNIT- IV

Motivation for generic programming – generic classes – generic methods – generic code and virtual machine – inheritance and generics – reflection and generics - Multi-threaded programming – interrupting threads – thread states – thread properties – thread synchronization – Executors – synchronizers.

JDBC: The connectivity Model, JDBC/ODBC Bridge, java.sql package, connectivity to remote database.

EXPERIMENTS

1. Basic programs of simple statements, conditional statements, iterative statements and arrays
2. Programs having object-oriented concepts like Inheritance and Interface
3. Programs for Exception Handling and Event Handling
4. Programs of Threads and Multithreading

5. Programs related to Applets and Swings
6. Programs including JAVA Beans and Servlets

Textbooks

1. Naughton, Schildt, The Complete Reference JAVA2, TMH.
2. Balaguruswamy E, Programming in JAVA, TMH
3. Cay S. Horstmann and Gary Cornell, "Core Java: Volume I – Fundamentals", Eighth Edition, Sun Microsystems Press
4. K. Arnold and J. Gosling, "The JAVA programming language", Third edition, Pearson Education, 2000.

Reference Books

1. Margaret Levine Young, The Complete Reference Internet, TMH.
2. Dustin R. Callway, Inside Servlets, Addison Wesley.
3. Mark Wutica, JAVA Enterprise Edition, QUE.
4. Steven Holzner, JAVA2 Black book, Dreamtech.
5. Timothy Budd, "Understanding Object-oriented programming with Java", Updated Edition, Pearson Education, 2000.
6. C. Thomas Wu, "An introduction to Object-oriented programming with Java", Fourth Edition, Tata McGraw-Hill Publishing company Ltd., 2006

MCA-202 DATA STRUCTURE & APPLICATIONS 5 Credits (3-1-2)

Course Objectives:

1. To understand the various techniques of sorting and searching
2. To design and implement arrays, stacks, queues, and linked lists
3. To understand the complex data structures such as trees and graphs

Learning Outcomes: On completion of this course, students are expected to be capable of understanding the data structures, their advantages and drawbacks, how to implement them in C, how their drawbacks can be overcome and what the applications are and where they can be used. Students should be able to learn about the data structures/ methods/algorithms mentioned in the course with a comparative perspective so as to make use of the most appropriate data structure/ method/algorithm in a program to enhance the efficiency (i.e. reduce the run-time) or for better memory utilization, based on the priority of the implementation.

UNIT- I: 9

Linear Data Structure Introduction: Need of data structure, Concepts a) Data and data structure b) Abstract Data Type and Data Type. Algorithms and programs, basic idea of pseudo-code, Algorithm efficiency and analysis, time and space analysis of algorithms – order notations Array: Different representations – row major, column major. Sparse matrix - its implementation and usage, Array representation of polynomials Linked List: Singly linked list, circular linked list, doubly linked list, linked list representation of polynomial and applications.

UNIT- II 9

Stack and Queue: Stack and its implementations (using array, using linked list), applications. Queue, circular queue, dequeue. Implementation of queue- both linear and circular (using array, using linked list), applications Recursion: Principles of recursion – use of stack, differences between recursion and iteration, tail recursion. Applications - The Tower of Hanoi, Eight Queens Puzzle

UNIT- III 9

Nonlinear Data structures Trees: Basic terminologies, forest, tree representation (using array, using linked list). Binary trees - binary tree traversal (pre-, in-, post- order), threaded binary tree (left, right, full) - non-recursive traversal algorithms using threaded binary tree, expression tree. Binary search tree- operations (creation, insertion, deletion, searching). Height balanced binary tree – AVL tree (insertion, deletion with examples only). B-Trees – operations (insertion, deletion with examples only). Graph definitions and concepts Graph representations/storage implementations – adjacency matrix, adjacency list, adjacency multi-list. Graph traversal and connectivity – Depth first search (DFS), Breadth-first search (BFS) – concepts of edges used in DFS and BFS, applications. Minimal spanning tree – Prim's algorithm (basic idea of greedy methods).

UNIT- IV 9

Searching, Sorting: Sorting Algorithms: Bubble sort and its optimizations, insertion sort, shell sort, selection sort, merge sort, quick sort, heap sort (concept of max heap, application – priority queue), radix sort. Searching: Sequential search, binary search, interpolation search Hashing: Hashing functions, collision resolution techniques.

EXPERIMENTS: Write C/C++ Programs to illustrate the concept of the following:

1. Arrays
2. Linked List
3. Stack
4. Queue
5. Graph
6. Tree
7. Searching & Sorting Algorithms

Books & References:

1. Data Structures and Program Design In C, 2/E by Robert L. Kruse, Bruce P. Leung.
2. Data Structures in C by Aaron M. Tenenbaum.
3. Fundamentals of Data Structures of C by Ellis Horowitz, Sartaj Sahni, Susan Anderson-freed.
4. Data Structures by S. Lipschutz.
5. Data Structures Using C by Reema Thareja.
6. Data Structure Using C, 2/e by A.K. Rath, A. K. Jagadev.
7. Introduction to Algorithms, by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein

MCA-203 **Web Technologies** **5 Credits**

Course Category : Department Core (DC)

Pre-requisite Subject : NIL

Contact Hours/Week : Lecture : 3, Tutorial : 1 , Practical: 2

Number of Credits : 5

Course AssessmentMethods : Continuous assessment through tutorials, attendance, homeassignments, quizzes, practical work, record, viva voce andThree Minor tests and One Major Theory & PracticalExamination

Course Outcomes : The students are expected to be able to demonstrate thefollowing knowledge, skills and attitudes after completing thiscourse

1. Identify common design mistakes when creating a web based application.
2. Discuss the process of editing a web page using text editors and web page editors
3. Cover commonly used HTML tags and discuss how this knowledge is important to aweb designer
4. Demonstrate an understanding of basic CSS, XML, JAVA Script, JSP, ASP.NET andPHP

Topics Covered

UNIT-I

Introduction to WWW- World Wide Web, WWW Architecture, Web Search Engines, WebCrawling, Web Indexing, Web Searching, Search Engines Optimization and Limitations, WebMining: Web Content Mining, Web Structure Mining, Web Usage Mining 9

UNIT-II

Markup Language Basics: SGML, HTML, CSS And XMLSGML: Standard Generalized Markup Language (SGML) -Structures, Elements, ContentModels, DTD, Attributes Entities.HTML: Designing Web Pages With HTML-Use Of Tags, Hyperlinks, URLs, Tables, TextFormatting, Graphics & Multimedia, Imagemap, Frames and Forms in Web Pages.CSS: Use of Cascading Style Sheet in Web Pages.XML: Extensible Markup Language (XML): Introduction using User-Defined Tags in WebPages, Displaying XML Contents, XML Dtds, Use of XSL

9

UNIT-III

Client-Side Scripting using JAVA ScriptJAVA script Overview; Constants, Variables, Operators, Expressions & Statements; UserDefined& Built-in Functions; Client-Side Form Validation; Using Properties and Methods ofBuilt-in Objects

9

UNIT-IV

Server-Side Scripting Using JSP, ASP.NET And PHPJSP :Introduction to JSP, JSP Architecture, JSP Directives, JSP Scripting Elements, DefaultObjects in JSP, JSP Actions, JSP with Beans and JSP with Database, Error Handling in JSP,Session Tracking Techniques in JSP, Introduction to Custom Tags.ASP.NET: ASP.Net

Coding Modules, ASP.NET Page Directives, Page Events and Page LifeCycle ,Postback and Crosspage Posting ASP.NET Server Controls , HTML Controls,Validation Controls, Building Databases.PHP(Hypertext Preprocessor)-Introduction, Syntax, Variables, Strings, Operators, If- Else,Loop, Switch, Array, Function, Form ,Mail, File Upload, Session, Error, Exception, Filter, PHPODBC 9

EXPERIMENTS

1. Create a HTML static web page which shows the use of different tags in that.
2. Insert an image and create a link such that clicking on image takes user to other page.
3. Prepare a sample code to illustrate three types of lists in HTML.
4. Use tables to provide layout to your HTML page describing your university infrastructure
5. Use frames such that page is divided into 3 frames 20% on left to show contents of pages, 60%in center to how body of page, remaining on right to show remarks.
6. Create a simple form that will show all the INPUT METHODS available in HTML.
7. Create a sample code to illustrate the Embedded, External and Inline style sheets for your webpage.
8. Write down simple JAVA Script using timeout such that image will be changed after every 1ms at a specified position.
9. Design a registration form and validate its field by using JAVA script.
10. Write an XML example of given tree that demonstrates the creation of user-designed tags anddisplay it in a browser.
11. college, employee, fname, lname, joindate, bdate, age, salary (with atleast 3 elements)
12. Write a program in XML for creation of DTD which specifies a particular set of rules.
13. Create a bean student with attributes (first name, last name, age, class). In another JSP pagedisplay the bean values using <jsp:usebean>.
14. Write a program to use JDBC connectivity program for maintaining database by sendingqueries through JSP Page.
15. Use ad-rotator to change advertisements on client side request.(ASP.NET)
16. Implement Session tracking using user authentication in ASP.NET.
17. Write a PHP script to create a database Student DB.
18. Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.

Textbooks

1. Uttam K. Roy, Web Technologies, 1/e, Oxford University Press, USA
2. M. Srinivasan, Web Technology: Theory and Practice, Pearson Education India
3. Deitel, Deitel and Nieto, Internet and Worldwide Web - How to Program, 5th Edition, PHI,2011.
4. Ralph Moseley & M. T. Savaliya , Developing Web Application- Second Edition, Wiley
5. Miller/Kirst, Web Programming Step by Step, Stepp, 2nd edition, 2009
6. Ullman , PHP for the Web: Visual Quick Start Guide, Pearson Education, 4th edition
7. www.w3c.org
8. www.w3schools.com

Readings:

Various journal and conference articles, research reports, and book excerpts as appropriate

Reference books

1. Ivan Bayross , Web Enabled Commercial Application Development Using HTML, DHTML,JAVA Script, Perl & CGI, BPB Publication, 2005
2. Hans Bergsten, JAVA Server Pages, O‟Reilly

MCA-204 OPERATING SYSTEM CONCEPTS 5 Credits (3-1-2)

Course Objectives:

This course should provide the students with good understanding of Operating System including its architecture and all its components. Good conceptions on all the subjects like processes, inter-process communication, semaphore, message passing, classical systems, security and protection mechanism, I/O hardware and software, deadlocks, etc.

Learning Outcomes:

1. List and define the fundamental concepts of Operating System
2. Study various CPU scheduling, Disk scheduling algorithms and compare their performance
3. Ability to simulate the given problems

UNIT- I **9**

Introduction: Basic architectural concepts, Operating System Services, interrupt handling, concepts of batch-processing, multiprogramming, time-sharing, real-time operations; Resource Manager view, process view and hierarchical view of an OS.

Memory management: Partitioning, paging, concepts of virtual memory demand paging, page replacement algorithms, segmentation, Segmentation and demand-paging, Cache memory management.

UNIT- II **9**

Processor management: CPU scheduling – short-term, medium term and long term scheduling, non-preemptive and preemptive algorithms, performance analysis of multiprogramming, multiprocessing and interactive systems, Concurrent processes,precedence graphs, critical section problem, semaphores; Classical process, co-ordination problems, Producer-consumer problem, Reader-writer problem, Dining philosophers problem, Barber's shop problem, Interprocess communication.

UNIT- III **9**

Concurrent Programming: Critical region, conditional critical region, monitors, Deadlocks: prevention, avoidance, detection and recovery. Device Management: Scheduling algorithms – FCFS, shortest-seek-time-first, SCAN, C- SCAN, LOOK, C-LOOK algorithms, spooling, spool management algorithm.

UNIT- IV **9**

Information Management: File concept, file support, directory structures, symbolic file directory, basic file directory, logical file system, physical file system, access methods, file protection, file allocation strategies. Protection: Goals, policies and mechanisms, domain of protection, access matrix and its implementation, access lists, capability lists, Lock/Key mechanisms, passwords, dynamic protection scheme, security concepts and public and private keys, RSA encryption and decryption algorithms.
A case study: A UNIX OS file system, shell, filters, shell programming, programming with the standard I/O, UNIX system calls.

Experiments

1. Study of hardware and software requirements of different operating systems(UNIX,LINUX,WINDOWS XP, WINDOWS7/8
2. Execute various UNIX system calls for
   a. Process management
   b. File management
   c. Input/output Systems calls
3. Implement CPU Scheduling Policies:
   a. SJ
   b. Priority
   c. FCFS
   d. Multi-level Queue
4. Implement file storage allocation technique:
   a. Contiguous(using array)
   b. Linked –list(using linked-list)
   c. Indirect allocation (indexing)
5. Implementation of contiguous allocation techniques:
   a. Worst-Fit
   b. Best- Fit
   c. First- Fit
6. Calculation of external and internal fragmentation
   a. Free space list of blocks from system
   b. List process file from the system
7. Implementation of compaction for the continually changing memory layout and calculate total movement of data
8. Implementation of resource allocation graph RAG)
9. Implementation of Banker‟s algorithm
10. Conversion of resource allocation graph (RAG) to wait for graph (WFG) for each type of method used for storing graph.
11. Implement the solution for Bounded Buffer (producer-consumer)problem using inter process communication techniques-Semaphores
12. Implement the solutions for Readers-Writer's problem using inter process communication technique - Semaphore

Books & References:

1. A. Silberschatz and P. B. Galvin: Operating Systems Concepts, 5th ed., John Wiley and Sons, New York, 1998.
2. J. L. Peterson and A. Silberschatz: Operating Systems Concepts, Addison-Wesley, Reading, Mass., 1987.
3. P. B. Hansen: Operating System Principles, Prentice Hall, Englewood Cliffs, 1980.
4. A. S. Tannenbaum: Modern Operating Systems, Prentice Hall, Englewood Cliffs, 1992.
5. S. E. Madnick and J. J. Donovan: Operating Systems, McGraw Hill, New York

MCA-205 Software Lab-2 2 Credits (0-1-2)

Course Objectives:

The objectives of this course are:

1. To give overview of advanced C programming (command line arguments & file handling)
2. To give knowledge about practical usage of LINUX system calls
3. To develop capability of system call programming for practical understanding of operating system's internal functioning

Learning Outcomes:

The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Understanding of process handling in Operating system
2. Usage of kernel system calls
3. Understanding of multiprogramming function of operating system using system calls

Experiments

1. Programs related to advanced C programming (command line arguments & file handling)
2. Understanding and usage of various LINUX kernel system calls
3. System call programming (programs based on LINUX kernel system calls using C language)

MCA-301 Artificial Intelligence Principles and Techniques 5 Credits 3-1-2

Course Objective

The primary objective of this course is to introduce the basic principles, techniques, and applications of Artificial Intelligence. Emphasis will be placed on the teaching of these fundamentals, not on providing a mastery of specific software tools or programming environments. Assigned projects promote a 'hands-on' approach for understanding, as well as a challenging avenue for exploration and creativity.

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

Course Outcomes

Upon successful completion of this course, the student shall be able to:

1) Demonstrate fundamental understanding of the history of artificial intelligence (AI) and its foundations.
2) Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.
3) Demonstrate awareness and a fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.

4) Demonstrate proficiency developing applications in an 'AI language', expert system shell, or data mining tool.
5) Demonstrate proficiency in applying scientific method to models of machine learning.
6) Demonstrate an ability to share in discussions of AI, its current scope and limitations, and societal implications

TOPICS TO BE COVERED

UNIT-I **09**

Introduction: The Foundations of Artificial Intelligence, The History of Artificial Intelligence Intelligent Agents, Agents and Environments, Good Behaviour, The Nature of Environments, The Structure of Agents, Solving Problems by Searching Problem-Solving Agents Searching for Solutions, Infrastructure for search algorithms, Measuring problem-solving performance. Uninformed Search Strategies, Informed (Heuristic) Search strategies, Greedy best-first search. A* search Heuristic Functions, Local Search Algorithms and Optimization Problem, Local Search in Continuous Spaces, Searching with Nondeterministic Actions, Online Search Agents and Unknown Environments.

UNIT-II **09**

Adversarial Search, Games, Optimal Decisions in Games, Alpha--Beta Pruning, Imperfect Real-Time Decisions, Stochastic Games, Partially Observable Games, State-of-the-Art Game Programs, Alternative Approaches, Defining Constraint Satisfaction Problems, Constraint Propagation: Inference in CSPs, Backtracking Search for CSPs, Variable and value ordering, Interleaving search and inference, Intelligent backtracking: Looking backward, Local Search for CSPs, The Structure of Problems, , reasoning, and planning Logical Agents Propositional vs. First-Order Inference Backward Chaining and Forward Chaining, Unification and Lifting.

UNIT-III **09**

Planning and Acting in the Real World, Definition of Classical Planning, Algorithms for Planning as State-Space Search, Planning Graphs, Classical planning as Boolean satisfiability, representing temporal and resource constraints, Planning and Acting in Nondeterministic Domains, Knowledge representation Techniques, Acting under Uncertainty, Probabilistic Reasoning, Time and Uncertainty.

UNIT-IV **09**

Different Forms of Learning, Supervised and Unsupervised Learning, Decision Trees Evaluating and Choosing the Best Hypothesis, A Logical Formulation of Learning, Statistical Learning with Complete Data, Natural Language Processing (Understanding Phase).

Books & References

1. S. Russel and P. Norvig, “Artificial Intelligence – A Modern Approach”, Second Edition, Pearson Education, 2012.

Reference Books

1. David Poole, Alan Mackworth, Randy Goebel, “Computational Intelligence: a logical approach”, Oxford University Press, 2012.
2. G. Luger, “Artificial Intelligence: Structures and Strategies for complex problem solving”, Fourth Edition, Pearson Education, 2012
3. J. Nilsson, “Artificial Intelligence: A new Synthesis”, Elsevier Publishers, 1998

MCA-302 Computer Network **5 Credits** **3-1-2**

Course Outcomes

The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Understand the concepts of communication architecture and protocols
2. Identify different types of communication mediums and techniques
3. Define and identify different types of multiplexing, data encoding, modulation, and switching techniques

4. Illustrate different standards of Local Area Network in terms of technologies and hardware used
5. Illustrate network addressing and analysis techniques
6. Understand the Wide Area Network technologies
7. Understand the network routing concepts
8. Understand the internetworking concepts and architectures
9. Understand the TCP/IP protocols and design architectures

TOPICS TO BE COVERED

UNIT-I **09**

Introductory Concepts: Goals and Applications of Networks, Network structure and architecture, the OSI reference model, services, networks topology, Physical Layer- transmission, switching methods, LAN Inter connection devices, Integrated services digital networks.

UNIT-II **09**

Medium access sub layer: Channel allocations, LAN protocols, ALOHA Protocols- Pure ALOHA, slotted ALOHA, Carrier Sense Multiple Access Protocols, CSMA with Collision Detection, Collision free Protocols, IEEE standards, Ethernet, FDDI, Data Link Layer- basic design issues, error correction & detection algorithms, elementary data link layer protocols, sliding window protocols, error handling, High Level Data Link Control

UNIT-III **09**

Network Layer: Packet switched networks – IP – ARP – RARP –DHCP – ICMP – Queuing discipline – Routing algorithms, congestion control algorithms, internetworking, TCP/IP protocol, IP addresses, IPv4 and IPv6.

UNIT-IV **09**

Transport Layer: Design issues, connection management, Internet Transport Protocol (UDP), Transmission Control Protocol. (TCP) -Adaptive Retransmission - Congestion control, Congestion avoidance –QoS.

Application Layer: Domain Name System, Electronic mail (Email), File Transfer Protocol, Hyper Text Transfer, Protocol, Introduction to Cryptography and Network Security (DES, RSA algorithms), Communication Security (IPSec, Firewalls).

EXPERIMENTS

1. To create scenario and study the performance of CSMA/CD protocol through simulation.
2. To create scenario and study the performance of token bus and token ring protocols through simulation.
3. Implementation of Error detection and correction algorithms.
4. Implementation and study of 1bit sliding window viz., stop and wait protocol.
5. Implementation and study of Go backN protocol.
6. Implementation and study of selective repeat protocol.
7. To get the MAC or Physical address of the system using Address Resolution Protocol.
8. Implementation of distance vector routing algorithm.
9. Implementation of link state routing algorithm.
10. To write a clientserver application for chat using TCP.
11. To write a C program to develop a DNS client server to resolve the given hostname.

Textbooks

1. Data Communication and Networking by Forouzan TMH
2. S Tanenbaum, Computer Networks, 4th, Edition", Pearson education

Reference Books

1. Data and Computer Communication by W. Stallings, Macmillan Press
2. Computer Networks with Internet Protocols by W Stallings, Pearson Education
3. Local and Metropolitan Area Networks by W Stallings, VIth edition, Pearson Education

MCA-303 Design and Analysis of Algorithms 5 Credits 3-1-2

Course Objective

The course introduces the basics of computational complexity analysis and various algorithm design paradigms. The goal is to provide students with solid foundations to deal with a wide variety of computational problems, and to provide a thorough knowledge of the most common algorithms and data structures.
The aim of this module is to learn how to develop efficient algorithms for simple computational tasks and reasoning about the correctness of them. Through the complexity measures, different range of behaviors of algorithms and the notion of tractable and intractable problems will be understood.

Course Outcomes

Students who complete the course will have the ability to demonstrate the following:

- **Knowledge and understanding**
  1. Understanding basic ideas about algorithms
  2. Understanding the concepts of time and space complexity, worst case, average case and best-case complexities and the big-O notation
  3. to apply knowledge of computing and mathematics to algorithm design
  4. Understanding the range of behaviours of algorithms and the notion of tractable and intractable problems
  5. Knowing and understanding a wide range of searching and sorting algorithms
  6. to analyse a problem and identify the computing requirements appropriate for its solution
  7. to design, implement, and evaluate an algorithm to meet desired needs
  8. to apply mathematical foundations, algorithmic principles, and computer science theory to the modelling and design of computer-based systems in a way that demonstrates comprehension of the trade-offs involved in design choices. - An ability to apply design and development principles in the construction of software systems of varying complexity.

• Cognitive skills (thinking and analysis).

1. Developing efficient algorithms for simple computational tasks - Reasoning about the correctness of algorithms
2. Computing complexity measures of algorithms, including recursive algorithms using recurrence relations

• Communication skills (personal and academic).

1. Ability to represent projects.

TOPICS TO BE COVERED

UNIT-I **09**

Introduction: Algorithms, Analysing Algorithms, Complexity of Algorithms, Growth of Functions, Performance Measurements, Sorting and Order Statistics
Sorting - Shell Sort, Quick Sort, Merge Sort, Heap Sort, Comparison of Sorting Algorithms, Sorting in Linear Time. Divide and Conquer Method with Examples such as Sorting, Matrix Multiplication, and Searching.

UNIT-II **09**

Greedy Methods with Examples such as Optimal Reliability, Resource Allocation, Knapsack, Minimum Spanning Trees – Prim‟s and Kruskal‟s Algorithms, Single Source Shortest Paths - Dijkstra‟s and Bellman Ford Algorithms.
Dynamic Programming with Examples such as Multistage Graphs, Knapsack, All Pair Shortest Paths - Warshal‟s and Floyd‟s Algorithms, Resource Allocation Problem.

UNIT-III **09**

Backtracking, Branch and Bound with Examples such as Travelling Salesman Problem, Graph colourings, N-Queen Problem, Hamiltonian Cycles and Sum of Subsets
Advanced Data Structures: Red-Black Trees, B – Trees, Binomial Heaps, Fibonacci Heaps.

UNIT-IV **09**

Selected Topics: String Matching, Text Processing- Justification of Text, Theory of NP- Complete and NP-Hard, Approximation Algorithms and Randomized Algorithms, Algebraic Computation.

EXPERIMENTS

1. To analyse time complexity of Insertion sort.
2. To analyse time complexity of Quick sort.
3. To analyse time complexity of Merge sort.
4. To Implement Largest Common Subsequence.
5. To Implement Matrix Chain Multiplication.
6. To Implement Strassen‟s matrix multiplication Algorithm, Merge sort and Quick sort.
7. To implement Knapsack Problem.
8. To implement Activity Selection Problem.
9. To implement Dijkstra‟s Algorithm.
10. To implement Warshall‟s Algorithm.
11. To implement Bellman Ford‟s Algorithm.
12. To implement Naïve String-Matching Algorithm.
13. To implement Rabin Karp String Matching Algorithm
14. To implement Prim‟s Algorithm.
15. To implement Kruskal‟s Algorithm

Textbooks

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, Introduction to Algorithms, PHI.
2. Ellis Horowitz and Sartaj Sahni, Fundamentals of Computer Algorithms, Computer Science Press, Maryland.

Reference Books

1. Knuth,D.E,FundamentalsofAlgorithms:TheArtofComputerProgramming.
2. Aho, Hopcraft,Ullman,“The Design and Analysis of Computer Algorithms”, Pearson Education.

MCA-304 Cloud Computing 5 Credits 3-1-2

Course Outcomes

The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Understand the concept of Existing Hosting Platforms and computing paradigms currently being used in industry and academia
2. Identify the issues related to Cloud Computing. To analyze IASS/ PAAS and SAAS services along with Cloud models.
3. Understand the concepts of various Cloud Platforms with comparative analysis and the concepts of virtualization with the advantages in Cloud.

TOPICS TO BE COVERED

UNIT-I **09**

Introduction: Basics of Emerging Cloud Computing Paradigm, Cloud Computing History and Evolution, Cloud Enabling Technologies, Practical Applications of Cloud Computing for Various Industries, Economics and Benefits of Cloud Computing Cloud Computing Architecture: Cloud Architecture Model, Types of Clouds:Public Private &Hybrid Clouds, Resource Management and Scheduling, QOS(Quality of Service) and Resource Allocation, Clustering

UNIT-II **09**

Classification of Cloud Implementations- Amazon Web Services - IaaS, Elastic Compute Cloud (EC2), Simple Storage Service (S3), Simple Queuing Services (SQS), VMware vCloud -IaaS, vCloud Express, Google AppEngine - PaaS, JAVA Runtime Environment

UNIT-III **09**

Data Center: Classic Data Centre, Virtualized Data Centre (Compute, Storage, Networking and Application), Business Continuity in VDC Virtualization: Virtualization, Advantages and disadvantages of Virtualization, Types of Virtualization: Resource Virtualization i.e. Server, Storage and Network virtualization, Migration of processes, VMware vCloud – IaaS

UNIT-IV **09**

Cloud Security and Privacy: Infrastructure Security: Infrastructure Security: The Network Level, Infrastructure

Security: The Host Level, Infrastructure Security: The Application Level, Data Security and Storage: Aspects of Data Security, Data Security Mitigation, Provider Data and Its Security. Privacy: DataLife Cycle, Key Privacy Concerns in the Cloud, Responsibility for Protecting Privacy, Changes to Privacy Risk Management and Compliance in Relation to Cloud Computing, Legal and Regulatory Implications

Textbooks

1. Dr. Kumar Saurabh, Cloud Computing, Wiley
2. ArshdeepBahga, Vijay Madisetti, Cloud Computing: A Hands-on Approach,Universities Press

Reference Books

1. Gerard Blokdijk, Ivanka Menken , The Complete Cornerstone Guide to CloudComputing BestPractices, Second Edition, Emereo Pty Ltd, 2009.
2. Anthony Velte, Toby Velte and Robert Elsenpeter, Cloud Computing: A practicalApproach,Tata McGraw Hil
3. Raj Kumar Buyya, James Broberg, Andrezei M. Goscinski, Cloud Computing:Principles andParadigms, , John Wiley and Sons2011.
4. Michael Miller, Cloud Computing, Pearson Education India, 2008.
5. Judith Hurwitz, Robin Bllor, Marcia Kaufmann, Fern Halper, Cloud Computingfor Dummies, Wiley, 2009

MCA-401 Internet of Things 5 Credits 3-1-2

Course Objectives:

Students will be explored to the interconnection and integration of the physical world and the cyber space. They will also be able to design &develop IoT devices.

Course Outcomes

On successful completion of the course, the student will:

- Understand the concepts of Internet of Things and its application areas
- Analyse the basic protocols in wireless sensor network and cloud
- Implement basic IoT applications on embedded platform
- Design IoT applications in different domains and be able to analyse their performance

TOPICS TO BE COVERED

UNIT-I **09**

Introduction to IoT: IoT Technology & Applications, Issues & Challenges, Integration of Sensors and Actuators, Sensor Networks, Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, Machine-to-Machine Communications, Difference between IoT and M2M.

UNIT-II **09**

Basics of Programming for developing IoT: Introduction to Arduino and Python programming. Implementation of IoT with Raspberry Pi :Introduction to Raspberry Pi, Raspberry Architecture, Raspberry OS & Programming, Raspberry Pi I/O Interfaces, Raspberry Communication Interfaces, Sensor based IoT application development on Raspberry Pi.

UNIT-III **09**

Data Management & Computing: Software Defined Networking, SDN for IoT, Network Function Virtualization, Interoperability in IoT, Cloud Computing, IoT Network & Cloud Services, Introduction to Cloud Service Model, Sensor-Cloud, Fog Computing

UNIT-IV **09**

Data Handling and Analytics, Bigdata management in IoT. Case Studies: Smart Cities, Smart Homes, Surveillance applications, Vehicular networks - Connected Vehicles, Smart Lighting System, Weather

Monitoring System, Smart Agriculture, Healthcare, Activity Monitoring, Industry applications, Other IoT applications.

EXPERIMENTS

Creating some IoT Project such as Led Blinking System, Push Button Control System for Light ON/OFF, Pattern Display System, LED Pattern with Push Button Control, 7 Segment Display System, Fire Alarm System, Remote Control System for Home Appliances like AC and Fan Regulator, Night Light Controlling& Monitoring System, Sensor Based Security System etc.

Books & References:

1. Pethuru Raj and Anupama C. Raman, "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press, Taylor & Francis Group, 2017, ISBN: 9781498761284
2. AdrianMcEwen, "Designing the Internet of Things", Wiley Publishers, 2013, ISBN: 978-1-118-43062-0
3. VijayMadisetti, ArshdeepBahga, "Internet of Things: A Hands-OnApproach",2014,ISBN:9780996025515
4. Daniel Kellmereit, "The Silent Intelligence: The Internet of Things", 2013, ISBN: 0989973700
5. WaltenegusDargie,ChristianPoellabauer, "Fundamentals of Wireless Sensor Networks: Theory and Practice", Wiley Publishers, 2010, ISBN 978-0-470-99765-9

MCA-351 Automata Theory 4 Credits 3-1-0

Course Objectives:

1. To give an overview of the theoretical foundations of computer science from the perspective of formal languages
2. To illustrate finite state machines to solve problems in computing
3. To explain the hierarchy of problems arising in the computer sciences.
4. To familiarize Regular grammars, context frees grammar.

Course Outcomes

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

1. To use basic concepts of formal languages of finite automata techniques
2. To Design Finite Automata's for different Regular Expressions and Languages
3. To Construct context free grammar for various languages
4. To solve various problems of applying normal form techniques, push down automata and
5. Turing Machines
6. To participate in GATE, PGECET and other competitive examinations

TOPICS TO BE COVERED

UNIT-I **09**

FINITE AUTOMATA (FA): Introduction, Deterministic Finite Automata (DFA) -Formal definition, simpler notations (state transition diagram, transition table), language of a DFA. Nondeterministic Finite Automata (NFA)- Definition of NFA, language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata, Finite automata with output (Moore and Mealy machines) and Inter conversion.

UNIT-II **09**

REGULAR EXPRESSIONS (RE): Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA's to Regular Expressions, Converting Regular Expressions to Automata, applications of Regular Expressions. REGULAR GRAMMARS: Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular -Pumping lemma, applications, Closure properties of regular languages.

UNIT-III **09**

CONTEXT FREE GRAMMER (CFG): Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFG's, Minimization of CFG's, CNF, GNF, Pumping Lemma for CFL's, Enumeration of Properties of CFL

UNIT-IV **09**

PUSHDOWN AUTOMATA: Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack and its Equivalence, Equivalence of CFG and PDA. TURING MACHINES (TM): Formal definition and behaviour, Languages of a TM, TM as accepters, Properties of recursive and recursively

enumerable languages, Universal Turing machine, The Halting problem, Undecidable problems about TMs.

Books:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2007), Introduction to Automata Theory Languages andComputation, 3rdedition, Pearson Education, India.

References:

1. K. L. P Mishra, N. Chandrashekaran (2003), Theory of Computer Science-AutomataLanguages and Computation, 2nd edition, Prentice Hall of India, India.

MCA-352 Python Programming 4 Credits 2-1-2

Course Outcomes

The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Write basic and advance Python programs
2. Write conditional and iterative statements in Python
3. Create arrays and use array methods in Python
4. Use various standard Python modules
5. Create functions and implement recursion in Python
6. Create and use Python classes and objects
7. Write code for Constructors, Destructors, Inheritance, Polymorphism and Exception handling
8. Write code for file handling and various file operations
9. Solve various real time problems using Python
10. Solve problems of Data Science and Machine Learning with Python

TOPICS TO BE COVERED

UNIT-I Programming Basics and Decision Making 09

Introduction: Key features and applications of Python, Python Editors and Compilers (Interpreters), Using different offline and online Python IDE, interacting with Python programs, Data types: Numeric, Boolean, Strings, Lists, Sets, Tuples, Dictionary; Variables: Declaration and initialization; Simple Statements: Taking inputs from user, displaying outputs, Other concepts: Operators, Expressions, Indentation, Comments, Casting; Conditional statements: If…Else

UNIT-II Control Flow and Other Programming Concepts 09

Iterative statements: For Loops, While Loops, Break, continue; Array: Looping Array elements, Array methods; Functions: Local and Global Variables, Built-in functions, User defined functions, Declaration of a function, Defining the function, Calling of the function, Functions with arguments, Recursion.

UNIT-III OOP and File Handling 09

Object Oriented Programming: Classes and objects, attributes and methods, constructors and destructors, inheritance, polymorphism, Exception Handling: Try…Except; Management of text files: Type of files, various file operations on text files, creating a text file, opening a file, closing a file, reading a text file, writing into a text file, copying a file to another file.

UNIT-IV Advance Concepts 09

Problem solving: Use of Python to solve real time problems, How Python helps to research problems, Creating various types of graphs corresponding to any data to show different kinds of results and analysis; Data Analysis: Understanding problems of data science and machine learning, Creating codes for data analysis problems in Python, Other advance programs

EXPERIMENTS

1. Writing codes using simple statements, operators and expressions
2. Writing codes using conditional statements
3. Writing codes using iterative statements
4. Writing programs for creating arrays, looping array elements and using array methods

5. Writing programs to use various standard modules
6. Writing codes to create functions and implement recursion
7. Writing object oriented codes for Constructors, Destructors, Inheritance, Polymorphism and Exception handling
8. Write codes for various file operations
9. Developing codes for solving various real time problems
10. Developing codes for solving problems of Data Science and Machine Learning
11. Writing codes to create various types of graphs corresponding to any data
12. Writing other advance programs in Python

Books & References:

1. Alex Martelli, "Python in a Nutshell"
2. Allen Downey, "Think Python"
3. Ken Lambert, "Fundamentals of Python: First Programs"
4. Willi Richert, Luis Pedro Coelho, "Building Machine Learning Systems with Python"
5. Cody Jackson ,"Learning to Program Using Python"
6. LjubomirPerkovic, "Introduction to Computing Using Python"
7. https://www.w3schools.com/python/default.asp
8. https://www.w3resource.com/python/python-tutorial.php
9. https://www.geeksforgeeks.org/python-tutorial/
10. https://www.geeksforgeeks.org/python-programming-language/

MCA-353 Information Security & Cyber Laws 4 Credits 3-1-0

Course Outcomes

The students are expected to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. list and define the fundamental concepts of Information Security and Cyber Laws.
2. manually solve a given (simple) Information Security problem to satisfy Cyber Laws.
3. manually infer the type of a given (simple) Information Security and Cyber Laws.
4. implement (simple) algorithms and data structures for Information Security and Cyber Laws.
5. design (large) solution for Information Security and Cyber Laws that are modular and have reusable components.
6. explain on a simple problem how Information Security and Cyber Laws are relevant.

TOPICS TO BE COVERED

UNIT-I **09**

History of Information Systems and its Importance, basics, Changing Nature of Information Systems, Need of Distributed Information Systems, Role of Internet and Web Services, Information System Threats and attacks, Classification of Threats and Assessing Damages, Security in Mobile and Wireless Computing, Security Challenges in Mobile Devices, authentication Service Security, Security Implication for organizations, Laptops Security, Concepts in Internet and World Wide Web: Brief review of Internet Protocols-TCP/IP, IPV4, IPV6. Functions of various networking components-routers, bridges, switches, hub, gateway, and Modulation Techniques

UNIT-II **09**

Basic Principles of Information Security, Confidentiality, Integrity Availability and other terms in Information Security, Information Classification, and their Roles. Security Threats to e-Commerce, Virtual Organization, Business Transactions on Web, e-Governance and EDI, Concepts in Electronics payment systems, E Cash, Credit/Debit Cards. Physical Security- Needs, Disaster and Controls, Basic Tenets of Physical Security and Physical Entry Controls, Access Control- Biometrics, Factors in Biometrics Systems, Benefits, Criteria for selection of biometrics, Design Issues in Biometric Systems, Interoperability Issues, Economic and Social Aspects, Legal Challenges Framework for Information Security, ISO 27001, SEE-CMM, Security Metrics, Information Security Vs Privacy

UNIT-III **09**

Model of Cryptographic Systems, Issues in Documents Security, System of Keys, Public Key Cryptography, Digital Signature, Requirement of Digital Signature System, Finger Prints, Firewalls, Design and Implementation Issues, Policies Network Security- Basic Concepts, Dimensions, Perimeter for Network Protection, Network Attacks, Need of Intrusion Monitoring and Detection, Intrusion Detection Virtual Private Networks- Need, Use of Tunneling with VPN, Authentication Mechanisms, Types of VPNs and their Usage, Security Concerns in VPN.

UNIT-IV **09**

Laws, Investigation and Ethics: Cyber Crime, Information Security and Law, Types & overview of Cyber Crimes, Cyber Law Issues in E-Business Management Overview of Indian IT Act, Ethical Issues in Intellectual property rights, Copy Right, Patents, Data privacy and protection, Domain Name, Software piracy, Plagiarism, Issues in ethical hacking.

Books & References:

1. Godbole, "Information Systems Security", Willey Publication
2. Merkov, Breithaupt, "Information Security", Pearson Education
3. Yadav, "Foundations of Information Technology", New Age, Delhi
4. Schou, Shoemaker, "Information Assurance for the Enterprise", Tata McGraw Hill
5. Sood, "Cyber Laws Simplified", Mc Graw Hill
6. Furnell, "Computer Insecurity", Springer
7. IT Act 2000

MCA-354 Android Programming **4 Credits** **2-1-2**

Course Outcomes

The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course.

1. Know the components and structure of mobile application development frameworks for Android.
2. Learn the basic and important design concepts and issues of development of mobile applications.
3. skill to develop mobile applications for android platform.

TOPICS TO BE COVERED

UNIT-I **09**

Introduction to Android: Overview, History, Features, Architecture, Anatomy of an Android Application, Creating first Android Application, Applying Styles, Linking Activities Using Intents, Returning Results from an Intent

UNIT-II **09**

Getting to know the android UI: Components of a Screen, Views and View Groups, Layouts, Display Orientation, Creating the User Interface Programmatically, Basic Views, Application using views, Gallery and Image View

UNIT-III **09**

Displaying pictures and menus with views: Menus with Views, options Menu, Saving and Loading, Data to Files, Storage Option, Using Static Resources, Creating and Using Databases, Application using menu

UNIT-IV **09**

Content providers: Sharing Data in Android, Query String Constants, Filtering, Sorting, Creating Content Providers, SMS Messaging, Application using intent for SMS, Application for broadcast receiver, Activity from a Broadcast Receiver, **Publishing App:** Tasks, communicating between a Service & an Activity, Deploying APK files.

Books:

1. Wei – Meng Lee, Beginning Android Application Development, Wiley publications.
2. Reto Meier, Professional Android 4 Application Development, Wiley publications

References:

1. Mark Murphy; Beginning Android 3; Apress Springer India Pvt Ltd. ;1st Edition; 2011;ISBN13: 978-1-4302-3297-1
2. Sayed Hashimi , Satya Komatineni, Dave MacLean; Pro Android 4; Apress Springer India Pvt Ltd; 1st Edition; 2012; ISBN: 978-1-4302-3930-7
3. The Android Developer's Cookbook: Building Applications with the Android SDK by James Steele, Nelson To, Addison-Wesley Professional

MCA-355 Distributed Database Systems **4 Credits** **3-1-0**

Course Outcomes

The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. Understand the key concepts and techniques of distributed database systems
2. Learn the concepts of semantics data control, query processing and query optimization
3. Expertise in transaction management and concurrency control techniques
4. Enhance the skills towards system reliability and recovery techniques.
5. Familiar with advanced distributed database concepts and techniques

TOPICS TO BE COVERED

UNIT-I **09**

Introduction: Distributed computing; What is a DDBS?; Advantages and disadvantages of DDBS; Problem areas; **Distributed Database Management System Architecture:** Transparencies in a distributed DBMS; ANSI/SPARC architecture; Alternatives in distributed database systems; Directory issues; Distributed database design;Distributed design issues; Fragmentation; Allocation alternatives.

UNIT-II **09**

Semantics Data Control: View management; Data security; Semantic Integrity Control; **Query Processing:** Objectives of query processing; Query processing components; Distributed query processing methodology; **Distributed Query Optimization:** Factors governing query optimization; Cost functions; Query optimization objectives; Ordering of fragment queries; Complexity of relational operations; Query optimization issues

UNIT-III **09**

Transaction Management: Transaction; Transaction properties; Goals of transaction management; Characteristics of transactions; Taxonomy of transaction models; **Concurrency Control:** Concurrency control in centralized database systems; Concurrency control in DDBSs; Concurrency control algorithms; Deadlock management

UNIT-IV **09**

Reliability: Logs; Faults; Failures; Types of failures; Reliability techniques; Reliability issues in DDBSs; Commit protocols; Recovery protocols; REDO Protocol; UNDO Protocol; Why Logging; Checkpointing; Shadowing; **Advanced Topics:** Mobile Databases; Distributed Object Management; Multi-databases; Basics of Cloud computing

Books & References:

1. M. Tamer Oezsu, Patrick Valduriez ``Principles of Distributed Database Systems, Second Edition'' Prentice Hall, 1999
2. Distributed Database Systems, D. Bell and J. Grimson, Addison-Wesley, 1992.
3. Distributed Systems: Concept and Design. Coulouris, Dollimore, and Kindberg. AW
4. Distributed Database Principles and Systems. Ceri and Pelagatti. McGraw Hill.
5. Recovery Mechanisms in Database Systems. Kumar and Hsu, Prentice Hall.
6. Concurrency Control and Recovery in Database Systems. Bernstein, Hadzilacos and Goodman, AW
7. Other materials required for the class will be made available during the course.

MCA-356 Computer Graphics **4 Credits** **3-1-0**

Course Outcomes

1. This course is designed to provide a comprehensive introduction to computer graphics leading to the ability to understand contemporary terminology, progress, issues, and trends.
2. A thorough introduction to computer graphics techniques, focusing on 3D modeling, image synthesis, and rendering.
3. Topics cover: geometric transformations, geometric algorithms, software systems (OpenGL, shades), 3D object models (surface, volume and implicit), visible surface algorithms, image synthesis.
4. Shading and mapping, ray tracing, global illumination, Monte Carlo path tracing, photon mapping, and anti-aliasing.
5. The interdisciplinary nature of computer graphics is emphasized in the wide variety of examples and applications.
6. Aiming at conducting Tutorial, Seminars and Remedial classes.

Course Outcomes

Students who complete the course will have the ability to demonstrate the following:

1. Students will demonstrate an understanding of contemporary graphics hardware.

2. Students will create interactive graphics applications in C++ using one or more graphics application programming interfaces.
3. Students will write program functions to implement graphics primitives.
4. Students will write programs that demonstrate geometrical transformations.
5. Students will demonstrate an understanding of the use of object hierarchy in graphics applications.
6. Students will write program functions to implement visibility detection.
7. Students will write programs that demonstrate computer graphics animation.
8. Students will write programs that demonstrate 2D image processing techniques.

TOPICS TO BE COVERED

UNIT-I **09**

Graphics Primitives: Display Devices: Refresh Cathode Ray Tube, Raster Scan Display, Plasma display, Liquid Crystal display, Plotters, Printers. Input Devices: Keyboard, Trackball, Joystick, Mouse, Light Pen, Tablet, and Digitizing Camera. Input Techniques: Positioning techniques, Positioning Constraints, Scales & Guidelines, Rubber-Band techniques, Dragging, dimensioning techniques and Graphical Potentiometers, Pointing and Selection: the use of selection points, defining a boundary rectangle, multiple selections, Menu selection.

Mathematics for Computer Graphics: Point representation, Vector representation, Matrices and operations related to matrices, Vector addition and vector multiplication, Scalar product of two vectors, Vector product of two vectors.

UNIT-II **09**

LineDrawing Algorithms: DDA algorithms, Bresenham's algorithm (Line, Circle, ellipse and etc).

Segment & Display files: Segments, Functions for segmenting the display file, Posting and unposting a segment, segment naming schemes, Default error conditions, appending to segments, Refresh concurrent with reconstruction, Free storage allocation, Display file Structure

Graphics Operations: Clipping: Point Clipping, Line Clipping. Polygon Clipping. Filling: Inside Tests, Flood fill algorithm, Boundary-Fill Algorithm and scan-line polygon fill algorithm

UNIT-III **09**

Conics, Curves and Surfaces: Quadric surfaces: Sphere, Ellipsoid, and Torus. Super quadrics: Super ellipse, super ellipsoid. Spline & Bezier Representations: Interpolation and approximation splines, parametric continuity conditions, Geometric Continuity Conditions, Spline specifications. Bezier curves and surfaces. Transformation: 2D transformation, Basic Transformations, Composite transformations: Reflection, Shearing, Transformation between coordinate systems. 3 D Graphics: 3 D Display Methods, 3 D modelling, 3 D transformations.

UNIT-IV **09**

Animation: Introduction to Animation, Principles of Animation, Types of Animation, Types of Animation Systems: Scripting, Procedural, Representational Stochastic.

Projections: Parallel projection, Perspective projection, Visible lines and surfaces identification.

Hidden Surface Removal: Need for Hidden Surface Removal, Depth-Buffer Algorithm, Properties thatHelpinReducingEfforts,ScanLineCoherenceAlgorithm,Span-Coherence Algorithm, Area-Coherence Algorithms, Warnock‟s Algorithm, Priority Algorithms

Books:

1. Hearn and Baker Computer Graphics with OpenGL and C, 3e, Prentice Hall

Reference books

1. Newman and Sproul, "Principle of Interactive Computer Graphics", McGraw Hill
2. Rogers, "Procedural Elements of Computer Graphics", McGraw Hill
3. Rogers and Adams, "Mathematical Elements of Computer Graphics", McGraw Hill
4. Foley James D, "Computer Graphics", AW Ed 2

MCA-357 Distributed Systems **4 Credits** **3-1-0**

Course Objectives:

1. To study the characteristics of OS for Multiprocessor and Multicomputer.
2. To learn the issues related to designing DOS.
3. To have a broad and up-to-date coverage of the principles and practice in the area of Distributed Systems.

4. To understand the heterogeneous systems such as computers, mobile phones, other devices and Internet) and their functionalities.

Course Outcomes

1. Knowledge about advanced concepts in OS
2. Developing skill set in developing a distributed system.
3. Designing and evaluation of algorithms and protocols for various distributed systems

TOPICS TO BE COVERED

UNIT-I **09**

Process Synchronization, Synchronization Mechanism, Process Deadlock, Architectural of Distributed system, Theoretical foundations: logical and vector clocks, causal ordering of messages, Chandy Lamport global state recording algorithms, cuts of distributed computation, termination detection. Distributed Mutual Exclusion: Classification of distributed mutual exclusion, requirement of mutual exclusion theorem, performance metric for distributed mutual exclusion algorithms

UNIT-II **09**

Distributed Deadlock Detection: deadlock handling strategies in distributed systems, Issues in deadlock detection & resolution, control organization for distributed dead lock detection, centralized dead lock detection algorithms, distributed dead lock detection algorithms, hierarchical dead lock detection algorithms.

Agreement Protocols: system model, classification of agreement problem, Solution to Byzantine Agreement problem, Application of Agreement algorithms.

UNIT-III **09**

Distributed Resource Management: distributed file system, mechanism for building distributed file systems, design issues, sun network file system, sprite file system, log-structured file system, disk space management, system, distributed shared memory: Algorithm for implementing DSM, Memory coherence, coherence protocols and design issues, Distributed Scheduling

UNIT-IV **09**

Failure recovery and Fault tolerance: backward and forward error recovery check pointing and recovery, recovery in concurrent systems, consistent set of checkpoints, synchronous check pointing and recovery, and asynchronous check pointing and recover. Fault tolerance: voting protocols, dynamic voting protocols, dynamic vote reassignment protocols.

EXPERIMENTS

1. Program to implement non token based algorithm for Mutual Exclusion.
2. Program to implement Lamport's Logical Clock.
3. Program to implement edge chasing distributed deadlock detection algorithm.
4. Program to implement locking algorithm.
5. Program to implement Remote Method Invocation.
6. Program to implement Remote Procedure Call.
7. Program to implement Chat Server.
8. Program to implement termination detection

Books:

1. Advanced Concept in Operating Systems-Singhal & Shivaratri (McGraw Hill)

References:

1. Distributed Operating Systems and Algorithm Analysis - Randy Chow & Theodore Johnson (Pearson Education)
2. Distributed System: Concepts and Design - Coulouris, Dollimore, Kindberg (Pearson Education)
3. Distributed Algorithms - Gerald Tel (Cambridge University Press)

MCA-358 Data Mining & Warehousing 4 Credits 3-1-0

Course Outcomes

The students are expected to be able to demonstrate the following knowledge, skills, and attitudes after completing this course.

1. Approach business problems data-analytically by identifying opportunities to derive business value from data.
2. know the basics of data mining techniques and how they can be applied to extract relevant business intelligence

TOPICS TO BE COVERED

UNIT-I **09**

Introduction to Data Mining: Motivation for Data Mining, Data Mining Definition &Functionalities, Classification of DM Systems, DM Task Primitives, Integration of a Data Mining System with A Database or A Data Warehouse, Major Issues in Data Mining. Data Warehousing. Overview of Concepts Like Star Schema, Fact and Dimension Tables, OLAP Operations, from OLAP to Data Mining. Data Pre-Processing:

Why? Descriptive Data Summarization, Data Cleaning: Missing Values, Noisy Data, Data Integration and Transformation. Data Reduction: Data Cube Aggregation, Dimensionality Reduction, Data Compression, Numerosity Reduction, Data Discretization and Concept Hierarchy Generation for Numerical and Categorical Data.

UNIT-II **09**

Mining Frequent Patterns, Associations and Correlations: Market Basket Analysis, Frequent Item Sets, Closed Item Sets, and Association Rules, Frequent Pattern Mining, Efficient and Scalable Frequent Item Set Mining Methods, The Apriori Algorithm for Finding Frequent Item Sets Using Candidate Generation, Generating Association Rules from Frequent Item Sets, Improving the Efficiency of Apriori, Frequent Item sets without Candidate Generation using FP Tree, Mining Multilevel Association Rules, Mining Multidimensional Association Rules, from Association Mining to Correlation Analysis, Constraint-Based Association Mining. Issues Regarding Classification and Prediction: Classification Methods: Decision Tree, Bayesian Classification, Rule Based Prediction: Linear and Non-Linear Regression Accuracy and Error Measures, Evaluating the Accuracy of a Classifier or Predictor.

UNIT-III **09**

Cluster Analysis: Types of Data in Cluster Analysis, Categories of Clustering Methods, Partitioning Methods K-Means, K-Medoids Hierarchical Clustering Agglomerative and Divisive Clustering, BIRCH and ROCK Methods, DBSCAN, Outlier Analysis Stream Data Classification, Clustering Association Mining in Stream Data. Mining Sequence Patterns in Transactional Databases.

UNIT-IV **09**

Spatial Data and Text Mining: Spatial Data Cube Construction and Spatial OLAP, Mining Spatial Association and Co-Location Patterns, Spatial Clustering Methods, Spatial Classification and Spatial Trend Analysis. Text Data Analysis and Information Retrieval, Dimensionality Reduction for Text, Text Mining Approaches Web Mining Introduction, Web Content Mining, Web Structure Mining, Web Usage Mining, Automatic Classification of Web Documents. Data Mining for Business Applications like Balanced Scorecard, Fraud Detection, Click Stream Mining, Market Segmentation, Retail Industry, Telecommunications Industry, Banking & Finance and CRM etc.

Books:

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 2nd Edition
2. P. N. Tan, M. Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Education

References:

1. MacLennan Jamie, Tang Zhao Hui and CrivatBogdan,Data Mining with Microsoft SQL Server 2008, Wiley India Edition.
2. G. Shmueli, N.R. Patel, P.C. Bruce, Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XL Miner, Wiley India.
3. Michael Berry and Gordon Linoff, Data Mining Techniques., 2nd Edition Wiley Publications
4. **Alex Berson and Smith, Data Mining and Data Warehousing and OLAP, McGraw Hill Publication.**
5. E. G. Mallach, Decision Support and Data Warehouse Systems", Tata McGraw Hill.
6. Michael Berry and Gordon Linoff, Mastering Data Mining- Art & science of CRM., Wiley Student Edition
7. Arijay Chaudhary & P. S. Deshpande, Multidimensional Data Analysis and Data Mining Dreamtech Press 8. Vikram Pudi & Radha Krishna, Data Mining, Oxford Higher Education.

MCA-451 Compiler Design **4 Credits** **2-1-2**

Course Objectives:

1. To teach concepts of language translation and phases of compiler design
2. To describe the common forms of parsers
3. To inculcate knowledge of parser by parsing LL parser and LR parser
4. To demonstrate intermediate code using technique of syntax directed translation
5. To Illustrate the various optimization techniques for designing various optimizing compilers

Course Outcomes

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

1. Use compiler construction tools and describes the Functionality of each stage of compilation process.
2. Construct Grammars for Natural Languages and find the Syntactical Errors/Semantic errors during the compilations using parsing techniques.
3. Analyse different representations of intermediate code.
4. Construct new compiler for new languages.
5. Participate in GATE, PGECET and other competitive examinations

TOPICS TO BE COVERED

UNIT-I **09**

INTRODUCTION TO COMPILERS: Definition of compiler, interpreter and its differences, the phases of a compiler, role of lexical analyzer, regular expressions, finite automata, from regular expressions to finite automata, pass and phases of translation, bootstrapping, LEX-lexical analyzer generator. PARSING: Parsing, role of parser, context free grammar, derivations, parse trees, ambiguity, elimination of left recursion, left factoring, eliminating ambiguity from dangling-else grammar, classes of parsing, top down parsing - backtracking, recursive descent parsing, predictive parsers, LL(1) grammars.

UNIT-II **09**

BOTTOM UP PARSING: Definition of bottom up parsing, handles, handle pruning, stack implementation of shift-reduce parsing, conflicts during shift-reduce parsing, LR grammars, LR parsers-simple LR, canonical LR(CLR) and Look Ahead LR (LALR) parsers, error recovery in parsing, parsing ambiguous grammars, YACC-automatic parser generator.

UNIT-III **09**

SYNTAX DIRECTED TRANSLATION: Syntax directed definition, construction of syntax trees, attributed and L-attributed definitions, translation schemes, emitting a translation. INTERMEDIATE CODE GENERATION: intermediate forms of source programs– abstract syntax
tree, polish notation and three address code, types of three address statements and its implementation, syntax directed translation into three-address code, translation of simple statements, Boolean expressions and flow-of-control statements.

UNIT-IV **09**

Type Checking, Run Time Environments, Code Optimization: Organization of code optimizer, basic blocks and flow graphs, optimization of basic blocks, the principal sources of optimization, the directed acyclic graph (DAG) representation of basic block, global data flow analysis.
CODE GENERATION: Machine dependent code generation, object code forms, the target machine, a simple code generator, register allocation and assignment, peephole optimization.

Books:

1. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman (2007), Compilers Principles, Techniques and Tools, 2nd edition, Pearson Education, New Delhi, India.

References:

1. Alfred V. Aho, Jeffrey D. Ullman (2001), Principles of compiler design, Indian student edition, Pearson Education, New Delhi, India.
2. Kenneth C. Louden (1997), Compiler Construction– Principles and Practice, 1st edition, PWS Publishing.
3. K. L. P Mishra, N. Chandrashekaran (2003), Theory of computer science- Automata Languages and computation, 2nd edition, Prentice Hall of India, New Delhi, India.
4. Andrew W. Appel (2004), Modern Compiler Implementation C, Cambridge University Press, UK.

MCA-452 Machine Learning 4 Credits 2-1-2

Course Objectives:

The course aims to provide an introduction to the basic principles, techniques, and applications of Machine Learning. The course covers the principles, design and implementation of learning programs that improve their performance on some set of tasks with experience. This one is organized primarily as a sequence of specific techniques, which comprise a small subset of the available machine learning algorithms. We will learn about details of these specific techniques and also use them to explore cross-cutting concepts

The goals of the course are

- To understand the basic building blocks and general principles that allow one to design machine learning algorithms
- To become familiar with specific, widely used machine learning algorithms.
- To learn methodology and tools to apply machine learning algorithms to real data and evaluate their performance

Course Outcomes

The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course:

1. To explain theory underlying machine learning
2. To construct algorithms to learn linear and non-linear models
3. To implement data clustering algorithms
4. To construct algorithms to learn tree and rule-based models

5. To apply reinforcement learning techniques

TOPICS TO BE COVERED

UNIT-I **09**

Foundations Of Learning: Components of Learning, Learning Models, Geometric Models, Probabilistic Models, Logic Models, Grouping and Grading, Learning Versus Design, Types of Learning, Supervised, Unsupervised, Reinforcement, Theory of Learning, Feasibility of Learning, Error and Noise, Training versus Testing, Theory of Generalization, Generalization Bound, Approximation- Generalization Trade-offs, Bias and Variance, Learning Curve

UNIT-II **09**

Linear Models: Linear Classification, Univariate Linear Regression, Multivariate Linear Regression, Regularized Regression, Logistic Regression, Perceptron, Multilayer Neural Networks, Learning Neural Networks Structures, Support Vector Machines, Soft Margin SVM, Going Beyond Linearity, Generalization and Over Fitting, Regularization, Validation

UNIT-III **09**

Distance Based Models: Nearest Neighbour Models, K-Means, Clustering around Medoids, Silhouettes, Hierarchical Clustering, K-D Trees, Locality Sensitive Hashing, Non-Parametric Regression, Ensemble Learning, Bagging and Random Forests, Boosting, Meta Learning

TREE BASED MODELS: - Decision Trees, Learning Decision Trees, Ranking and Probability Estimation Trees, Regression Trees, Clustering Trees

UNIT-IV **09**

Rule Models: Learning Ordered Rule Lists, Learning Unordered Rule Lists, Descriptive Rule Learning, Association Rule Mining, First-Order Rule Learning

Reinforcement Learning: Passive Reinforcement Learning, Direct Utility Estimation, Adaptive Dynamic Programming, Temporal-Difference Learning, Active Reinforcement Learning, Exploration, Learning an Action, Utility Function, Generalization in Reinforcement Learning, Policy Search, Applications in Game Playing, Applications in Robot Control

EXPERIMENTS

1. A simple *linear regression* attempts to draw a straight line that will best minimize the residual sum of squares between the observations and the predictions in python program language
2. Linear Regression Logistic Regression used in python program language
3. Decision Tree in python program language
4. SVM used in python program language
5. Naive Bayes used in python program language
6. KNN in python program language
7. K-Means in python program language
8. Random Forest in python program language
9. Dimensionality Reduction Algorithms in python program language
10. Gradient Boost &Ada boost in python program language

Books:

1. EthemAlpaydın -Introduction to Machine Learning Third Edition, MIT Press, 2004

References:

1. Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, Learning from Data, AML Book Publishers, 2012.
2. P. Flach, Machine Learning: The art and science of algorithms that make sense of data, Cambridge University Press, 2012.
3. K. P. Murphy, Machine Learning: A probabilistic perspective, MIT Press, 2012.
4. C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2007.
5. D. Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012.
6. M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning, MIT Press, 2012.
7. T. M. Mitchell, Machine Learning, McGraw Hill, 1997.
8. S. Russel and P. Norvig, Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall, 2009.

MCA-453 Wireless Sensor Networks 4 Credits 3-1-0

Course Objectives:

Students will be able to design & develop WSN applications for the real-world problems. They will also explore the interconnection and integration of the physical world and the cyber space.

Course Outcomes

On successful completion of the course, the student will:

1. Understand the concepts of wireless sensor networks and its application areas
2. Analyze the basic protocols in wireless sensor network
3. Implement basic WSN applications
4. Design WSN applications in different domains and be able to analyze their performance

TOPICS TO BE COVERED

UNIT-I: AD HOC NETWORKS – INTRODUCTION AND ROUTING PROTOCOLS 09

Basics of WSN: Basic components of a sensor node, Types of sensors, Constraints on the sensor nodes, WSN & its application areas, characteristics of WSN, Nature of Data in Sensor Networks, Manual vs Randomized node deployment, Event aware topology management in WSN, Issues & challenges with WSN, WSN coverage and placement, Localization and Positioning, Task driven sensing, Data Acquisition, Data Dissemination, Aggregation, Mobile WSN, Virtual Sensor Network, Operating Systems for WSN

UNIT-II: SENSOR NETWORKS – INTRODUCTION & ARCHITECTURES 09

MAC Protocols: Fundamentals of MAC Protocols, Design Issues, Overview of IEEE 802.15.4 and ZigBee, Contention-Free Medium Access, Contention-Based Medium Access, MAC Protocols for WSN: Contention-Free MAC Protocols, Contention-Based MAC Protocols, Hybrid MAC Protocols, Characteristics of MAC Protocols in Sensor Networks

UNIT-III :WSN NETWORKING CONCEPTS AND PROTOCOLS 09

Routing Protocols: Classification of routing protocols, Proactive routing vs Reactive routing, QoS routing, Flat Protocols: SPIN (Sensor Protocols for Information via Negotiation), Directed Diffusion, Hierarchical or Cluster Based Protocols: LEACH (Low Energy Adaptive Clustering Hierarchy), PEGASIS (Power-Efficient Gathering in Sensor Information Systems), Location Based Protocols:GEAR(Geographic and Energy Aware Routing), Some Other Protocols

UNIT-IV: SENSOR NETWORK SECURITY 09

Sensor Network Applications Case Studies: Military Applications, Environmental monitoring applications, Traffic Monitoring, Weather Monitoring, Fire Detection, Underwater Monitoring, Underground Monitoring, Agricultural Applications, Habitat Monitoring, IoT related applications, other applications

Books & References:

1. WaltenegusDargie,ChristianPoellabauer, "Fundamentals of Wireless Sensor Networks: Theory and Practice", Wiley Publishers, 2010, ISBN: 978-0-470-99765-9
2. Carlos De Morais Cordeiro, Dharma Prakash Agrawal, “Ad Hoc and Sensor Networks: Theory and Applications”, World Scientific Publishers, 2011, ISBN: 981-256-681-3
3. Dorothea Wagner and Roger Wattenhofer, “Algorithms for Sensor and Ad Hoc Networks”, Advanced Lectures, Springer, Lecture Notes in Computer Science 4621, 2007, ISBN-13 978-3-540-74990-5

MCA-454 Mobile Computing 4 Credits 3-1-0

Course Outcomes

The students are expected to be able to demonstrate the following knowledge, skills and attitudes after completing this course

1. To study the working principles of wireless LAN and its standards.
2. Demonstrate the energy management in wireless mobile networks.
3. Outline knowledge on Mobile IP.
4. Be familiar with the network protocol stack
5. Learn the basics of mobile telecommunication system
6. Be exposed to Ad-Hoc networks
7. Gain knowledge about different mobile platforms and application development
8. To build skills in working with Wireless application Protocols to develop mobile content applications

TOPICS TO BE COVERED

UNIT-I 09

Introduction to Electromagnetic Spectrum, modulation techniques, Mobile telephone systems, Cellular systems

development and GSM/CDMA Standards, handover scenarios, HSCSD and GPRS.

UNIT-II **09**

Satellite Systems-GEO, LEO, MEO, Broadcast Systems-Broadcast transmission, Digital Audio Broadcasting-Multimedia Object Transfer Protocol. Digital Video Broadcasting. Infrastructure and ad hoc networks, 802.11-Bluetooth- Architecture, Applications and Protocol, Layers, Frame structure, comparison between 802.11 and 802.16. Wireless ATM- Services, Reference Model, Functions, Radio Access Layer. Handover- Reference Model, Requirements, Types,

Location Management, Addressing, Access Point Control Protocol (APCP).

UNIT-III **09**

Mobile Network and Transport Layers: Mobile IP- Goals, Requirements, IP packet delivery, Advertisement, and discovery. Registration, Tunnelling and Encapsulation, Optimization, Reverse Tunnelling, IPv6, Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, Transmission.

UNIT-IV **09**

Dynamic Host configuring protocol, Ad-hoc networks – Routing, DSDV, Dynamic source routing. Hierarchical Algorithms. Wireless Application Protocol & World Wide Web: WAP-Architecture, Protocols-Datagram, Transaction, Session -Wireless Application Environment-WML- Features, Script- Wireless Telephony Application. WWW- HTTP, Usage of HTML, WWW system architecture.

Books:

1. Jochen Schiller, Mobile Communications, Preason Education Asia.
2. HazysztofWesolowshi, “Mobile Communication Systems”, John Wiley and Sons Ltd, 2002.

References:

1. Leon-Garcia & Indra Widjaja, Communication Networks -Fundamental Concepts and Key Architectures, Tata McGraw Hill.
2. Mobile Computing, ASOKE TALUKDER HASAN AHMED ROOPA R YAVAGAL,SecondEdition.McGrawHill

MCA-455 Big Data Technologies 4 Credits 3-1-0

Course Outcomes

After completing this course, the student will be able to

1. Demonstrate knowledge of Big Data Analytics concepts and its applications in business.
2. Demonstrate functions and components of Map Reduce Framework and HDFS.
3. Discuss Data Management concepts in NoSQL environment.
4. Explain process of developing Map Reduce based distributed processing applications.
5. Explain process of developing applications using HBASE, Hive, Pig etc.

TOPICS TO BE COVERED

UNIT-I **09**

Introduction to Big Data: Types of digital data, history of Big Data innovation, introduction to Big Data platform, drivers for Big Data, Big Data architecture and characteristics, 5 Vs of Big Data, Big Data technology components, Big Data importance and applications, Big Data features – security, compliance, auditing and protection, Big Data privacy and ethics, Big Data Analytics, Challenges of conventional systems, intelligent data analysis, nature of data, analytic processes and tools, analysis vs reporting, modern data analytic tools

UNIT-II **09**

Hadoop: History of Hadoop, Apache Hadoop, the Hadoop Distributed File System, components of Hadoop, data format, analysing data with Hadoop, scaling out, Hadoop streaming, Hadoop pipes, Hadoop Echo System, **Map Reduce:** Map Reduce framework and basics, how Map Reduce works, developing a Map Reduce application, unit tests with MR unit, test data and local tests, anatomy of a Map Reduce job run, failures, job scheduling, shuffle and sort, task execution, Map Reduce types, input formats, output formats, Map Reduce features, Real-world Map Reduce

UNIT-III **09**

HDFS (Hadoop Distributed File System): Design of HDFS, HDFS concepts, benefits and challenges, file sizes, block sizes and block abstraction in HDFS, data replication, how does HDFS store, read, and write files, Java interfaces to HDFS, command line interface, Hadoop file system interfaces, data flow, data ingest with Flume and Scoop, Hadoop archives, Hadoop I/O: compression, serialization, Avro, and file-based data

structures, **Hadoop Environment:** Setting up a Hadoop cluster, cluster specification, cluster setup and installation, Hadoop configuration, security in Hadoop, administering Hadoop, HDFS monitoring & maintenance, Hadoop benchmarks, Hadoop in the cloud

UNIT-IV **09**

NoSQL Databases: Introduction to NoSQL, **MongoDB:** Introduction, data types, creating, updating and deleing documents, querying, introduction to indexing, capped collections, **Spark:** Installing spark, spark applications, jobs, stages and tasks, Resilient Distributed databases, anatomy of a Spark job run, Spark on YARN, **SCALA:** Introduction, classes and objects, basic types and operators, built-in control structures, functions and closures, inheritance, **Hadoop Eco System Frameworks**: Applications on Big Data using Pig, Hive and HBase, **PIG**- Introduction to PIG, Execution Modes of Pig, Comparison of Pig with Databases, Grunt, Pig Latin, User Defined Functions, Data Processing operators, **Hive** - Apache Hive architecture and installation, Hive shell, Hive services, comparison with traditional databases, HiveQL, tables, querying data and user defined functions, sorting and aggregating, Map Reduce scripts, joins & subqueries, **HBase** – Hbase concepts, clients, example, Hbase vs RDBMS, advanced usage, schema design, advance indexing, Zookeeper – how it helps in monitoring a cluster, how to build applications with Zookeeper.

Books & References:

1. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging BusinessIntelligence and Analytic Trends for Today's Businesses", Wiley
2. Big-Data Black Book, DT Editorial Services, Wiley
3. Dirk deRoos, Chris Eaton, George Lapis, Paul Zikopoulos, Tom Deutsch, “Understanding Big Data Analytics forEnterprise Class Hadoop and Streaming Data”, McGrawHill.
4. Thomas Erl, Wajid Khattak, Paul Buhler, “Big Data Fundamentals: Concepts, Drivers and Techniques”, PrenticeHall.
5. Bart Baesens “Analytics in a Big Data World: The Essential Guide to Data Science and its Applications (WILEYBig Data Series)”, John Wiley & Sons
6. ArshdeepBahga, Vijay Madisetti, “Big Data Science & Analytics: A HandsOn Approach
7. Anand Rajaraman and Jeffrey David Ullman, “Mining of Massive Datasets”, CUP
8. Tom White, "Hadoop: The Definitive Guide", O'Reilly.
9. Eric Sammer, "Hadoop Operations", O'Reilly.
10. Chuck Lam, “Hadoop in Action”, MANNING Publishers
11. Deepak Vohra, “Practical Hadoop Ecosystem: A Definitive Guide to Hadoop-Related Frameworks and Tools”,Apress
12. E. Capriolo, D. Wampler, and J. Rutherglen, "Programming Hive", O'Reilly
13. Lars George, "HBase: The Definitive Guide", O'Reilly.
14. Alan Gates, "Programming Pig", O'Reilly.
15. Michael Berthold, David J. Hand, “Intelligent Data Analysis”, Springer
16. Bill Franks, “Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics

MCA-456 Data Sciences & Analysis **4 Credits** **3-1-0**

Course Outcomes:

TOPICS TO BE COVERED

UNIT-I **09**

Introduction to Data Science – Evolution of Data Science – Data Science Roles – Stages in aData Science Project – Applications of Data Science in various fields – Data Security Issues.Data operations: Reading, selecting, filtering,manipulating, sorting, grouping, rearranging, ranking, and plotting.

UNIT-II **09**

Data Collection Strategies – Data Pre-Processing Overview – Data Cleaning – DataIntegration and Transformation – Data Reduction – Data Discretization.Descriptive Statistics – Mean, Standard Deviation, Skewness and Kurtosis – Box Plots –Pivot Table – Heat Map – Correlation Statistics – ANOVA.

UNIT-III **09**

Simple and Multiple Regression – Model Evaluation using Visualization – Residual Plot – Distribution Plot – Polynomial Regression and Pipelines – Measures for In-sample Evaluation – Prediction and Decision Making.

UNIT-IV **09**

Generalization Error – Out-of-Sample Evaluation Metrics – Cross Validation – Overfitting –Under Fitting and Model Selection – Prediction by using Ridge Regression – TestingMultiple Parameters by using Grid Search.

Books & References:

1. JojoMoolayil, "Smarter Decisions: The Intersection of IoT and Data Science",PACKT, 2016.
2. Cathy O'Neil and Rachel Schutt, "Doing Data Science", O'Reilly, 2015.
3. David Dietrich, Barry Heller, Beibei Yang, "Data Science and Big data Analytics",EMC 2013
4. Raj, Pethuru, "Handbook of Research on Cloud Infrastructures for Big DataAnalytics", IGI Global.

MCA-457 Natural Language Processing 4 Credits 2-1-2

Course Objectives:

1. To learn the fundamentals of natural language processing
2. To understand the use of CFG and PCFG in NLP
3. To understand the role of semantics of sentences and pragmatics
4. To apply the NLP techniques to IR applications

Course Outcomes

Students who complete the course will have the ability to demonstrate the following:

1. To tag a given text with basic Language features
2. To design an innovative application using NLP components
3. To implement a rule-based system to tackle morphology/syntax of a language
4. To design a tag set to be used for statistical processing for real-time applications
5. To compare and contrast the use of different statistical approaches for different types of NLP applications.
6. Design, implement and test algorithms for NLP problems (measured by problem sets)

TOPICS TO BE COVERED

UNIT-I **09**

Introduction to Natural Language Understanding: Motivations,The study of language, Evaluating language understanding, Different levels of language analysis, Organization of Natural Language understanding system.
Regular Expressions: Regular Expressions, Patterns and their limitations, Finite-state automata-Regular and context Free Language. Practical regular expressions for finding and counting language phenomena.
Words and Word Forms: Morphology Paradigms- Inflectional Morphology, Derivational Morphology, Finite State Machine Based Morphological Parsing, Automatic Morphology Learning, Named Entities, Maximum Entropy Models
Linguistic Backgrounds: An outline of English syntax, Semantics and Pragmatics.

UNIT-II **09**

N-gram Models of syntax, counting words, Unsmoothed N-grams, Smoothing Backoff, Deleted Interpolation, Entropy, English word classes.
Part of Speech Tagging and Hidden Markov Models- Rule Based Part of Speech Tagging, Stochastic Part of Speech Tagging, Transformation-Based Part of Speech Tagging.

UNIT-III **09**

Knowledge Representation: Propositional Logic, First-order Logic, Rule-based System, Semantic Networks, Frames, Script, Challenges/Issues in Knowledge Representation

Grammars and Parsing: Grammar and Sentence structure, Top-Down and Bottom-Up Parsing, Shallow Parsing, Statistical parsing and probabilistic CFGs (PCFGs), Lexicalized PCFGs

UNIT-IV **09**

Ambiguity Resolution: Statistical Methods, Probabilistic Language Processing, Estimating Probabilities, Lexical Probabilities, Best First Parsing, Semantics and Logical Forms, Lexical semantics and word-sense disambiguation. Compositional semantics. Semantic Role Labeling and Semantic Parsing, Scope Ambiguity and Attachment Ambiguity resolution.

Applications: Information Extraction-. Named entity recognition and relation extraction, Sentiment Analysis; Machine Translation - Basic issues in MT, Statistical Translation, Word Alignment, Phrase-Based Translation, and Synchronous Grammars.

EXPERIMENTS

1. Write a program to generate tokens.
2. Write a program Part of Speech Tagging.
3. Write a program to test the syntax of the given sentence on a given grammar.
4. Write a program to represent knowledge.

NLP Projects for beginners

1. **Sentiment analysis for marketing**

This type of project can show you what it's like to work as an NLP specialist. For this project, you want to find out how customers evaluate competitor products, i.e. what they like and dislike. It's a great business case. Learning what customers like about competing products can be a great way to improve your own product, so this is something that many companies are actively trying to do.

To achieve this task, you will employ different NLP methods to get a deeper understanding of customer feedback and opinion.

2. **Toxic comment classification**

In this project, you want to create a model that predicts to classify comments into different categories. Comments in social media are often abusive and insulting. Organizations often want to ensure that conversations don't get too negative. This project was a Kaggle challenge, where the participants had to suggest a solution for classifying toxic comments in several categories using NLP methods.

3. **Create text summarizer**

Text summarization is one of the most interesting problems in NLP. It's hard for us, as humans, to manually extract the summary of a large document of text.

To solve this problem, we use automatic text summarization. It's a way of identifying meaningful information in a document and summarizing it while conserving the overall meaning.

The purpose is to present a shorter version of the original text while preserving the semantics.

In this project, you could use different traditional and advanced methods to implement automatic text summarization, and then compare the results of each method to conclude which is the best to use for your corpus.

4. **Translate and summarize news**

You can build a web app that translates news from Hindi to English or English to Hindi and summarizes them, using great Python libraries like newspaper, transformers, andgradio.

Where:

- Newspaper3k (11.1k stars): scrape almost any news website
- HuggingFace Transformers (48k): use state-of-the-art natural language models
- Gradio (2.9k): build interactive web-based demos

Books:

1. Daniel Jurafsky, James H. Martin―Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, Pearson Publication, 2014.
2. Steven Bird, Ewan Klein and Edward Loper, ―Natural Language Processing with Python, First Edition, OReilly Media, 2009.
3. Allen, James. Natural Language Understanding. The Benjamin/Cummings Publishing Company, Inc., Redwood City, CA.

References:

1. Steven Bird, Ewan Klein and Edward Loper, ―Natural Language Processing with Python, First Edition, OReilly Media, 2009.
2. Tanveer Siddiqui, U.S. Tiwary, ―Natural Language Processing and Information Retrieval, Oxford University Press, 2008

MCA-458 **Block chain & Cryptocurrency** **4 Credits** **3-1-0**

Course Outcomes

After the completion of this course, student will be able to:

1. Understand and explore the working of Blockchain technology (Understanding)
2. Analyse the working of Smart Contracts (Analyse)
3. Understand and analyse the working of Hyperledger (Analyse).
4. Apply the learning of solidity and de-centralized apps on Ethereum (Apply).

TOPICS TO BE COVERED

UNIT-I **09**

Introduction of Cryptography and Blockchain: What is Blockchain, Blockchain Technology Mechanisms & Networks, Blockchain Origins, Objective of Blockchain, Blockchain Challenges, Transactions and Blocks, P2P Systems, Keys as Identity, Digital Signatures, Hashing, and public key cryptosystems, private vs. public Blockchain, Blockchain Applications: Internet of Things, Medical Record Management System, Domain Name Service and Future of Blockchain.

UNIT-II **09**

Cryptocurrency: History, Distributed Ledger, Bitcoin protocols - Mining strategy and rewards, Ethereum - Construction, Smart Contract, Vulnerability, Attacks, Cryptocurrency Regulation: Stakeholders, Roots of Bit coin, Legal Aspects-Crypto currency Exchange, Black Market and Global Economy Bitcoin: The Bitcoin Network, The Bitcoin Mining Process, Mining Developments, Bitcoin Wallets, Decentralization and Hard Forks, Ethereum Virtual Machine (EVM), Merkle Tree, Double-Spend Problem, Blockchain and Digital Currency, Transactional Blocks, Impact of Blockchain Technology on Cryptocurrency.

UNIT-III **09**

Introduction to Cyber security, Need of cyber security, Malware & its types: Adware, Spyware, Virus, Worms, Trojan-horse, Scareware, Browser hacking software, Cyber crime and its kinds: Cyber Stalking, Child pornography, Forgery & counterfeiting, Software piracy & crimes related to IPRs, Cyber terrorism, Phishing, Computer Vandalism, Computer Hacking, Spamming, Cross site scripting, Online auction fraud, Cyber-squatting, Logic Bombs, Internet time theft, Deniel of service attack, salami attack, Data diddling, Email spoofing.

UNIT-IV **09**

Counter Cyber Security Measures: Authentication, Encryption, Digital Signature, Anti-Virus, Firewall, Steganography, Computer Forensics, Generating secure Passwords, Enabling two-step verification, securing computer using free anti-virus, Safe browsing guidelines for social networking sites: Tips for using social networking sites safely, posting personal details, friends, followers and contacts, status updates, sharing online contents, Revealing your location, sharing videos and photos, instant chats, joining and creating groups, Events & Communities, Email security tips.

Books & References:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press (July 19, 2016).
2. Antonopoulos, Mastering Bitcoin.
3. D. Drescher, Blockchain Basics. A press, 2017.
4. Introduction to Cybersecurity by Jeetendra Pande, Uttarakhand Open University, Haldwani
5. Cybersecurity by Neena Godbole, SunitBelapore, Wiley Publication