5 Years Integrated MSc (Computer Science)

[Course Initiated in 2021-2023]

Semester - IV

Scheme & Course Structure

Applicable from January, 2023

Course Structure with Credits

	Course Type	Subject Names & Credits	Total Credits
Semester-IV	Core Courses	• MCSCC221-Structured and Object Oriented Analysis and Design (4) • MCSCC222-Data Structures (4) • MCSCC223- Web Application Development (4) • MCSCC224- Data Structures – Practical (2) • MCSCC225- Web Application Development – Practical (2)	**16**
	Ability Enhancement Courses	• MCSAE221- Technical Writing & Communication Skills (2)	**2**
	Skill Enhancement Courses	• MCSSE221- Project – 2 (4)	**4**
	Discipline Specific Elective Courses		**—**
	Generic Elective Courses	• MCSGE221 - Programming with Python (2)	**2**
	Semester-IV (Total Credits)		**24**

TEACHING & EXAMINATION SCHEME FOR 5 YEAR INTEGRATED M. Sc. (COMPUTER SCIENCE) COURSE

Semester-IV	Course		Hours per week								Credits
	Course Code	Name	Theory	Practical	Internal	Total	Lectures	Others (Tutorials)	Practical	Total	
Semester-IV	MCSCC221	Structured and Object Oriented Analysis and Design	70	***	30	100	3	1	***	4	4
	MCSCC222	Data Structures	70	***	30	100	3	1	***	4	4
	MCSCC223	Web Application Development	70	***	30	100	4	***	***	4	4
	MCSCC224	Data Structures – Practical	***	35	15	50	***	***	***	4	2
	MCSCC225	Web Application Development – Practical	***	35	15	50	***	***	4	4	2
	MCSSE221	Project – 2	***	70	30	100	***	***	8	8	4
	MCSAE221	Technical Writing & Communication Skills	35	***	15	50	2	***	***	2	2
	MCSGE221	Programming with Python	***	35	15	50	***	***	4	4	2
		Total	245	175	180	600	12	2	16	34	24

Theory/Tutorial Credits → 1 Hour = 1 Credit

Practical Credits → 2 Hours = 1 Credit

Course Name: Structured and Object Oriented Analysis and Design

Course Code: MCSCC221

Objectives:

The aim of this course is to enable students to

- Develop proficiency in the core concepts, methodologies and principles of software analysis and design
- Get a good understanding of major approaches like structured design and object oriented design
- Apply various tools for analysis and design of software

Prerequisites:

Fundamentals of structured programming and object oriented programming

Contents:

1. **Structured Analysis and Design**
   Types of systems, role of system analyst, system development life cycle, CASE tools, interviewing, Joint Application Development, Questionnaires, Data flow approach, developing data flow diagrams, logical and physical data flow diagram, data dictionary, process specification, structured English, Decision table, Decision tree

2. **Object Modelling concepts**
   Introduction, Modelling as a design technique, class modelling-object and classes, association, generalization, aggregation, abstract class, multiple inheritance, metadata, reification, constraints, derived data, packages, state modelling-state, transitions and conditions, state diagram, nested state diagram, nested states, signal generalization, concurrency

3. **Class and Interaction modelling**
   Classes, relationships, common mechanisms, class diagrams, interactions, use cases, use case diagram, interaction diagram, activity diagram, and sequence diagrams

4. **Object Oriented Analysis and Design**
   Process overview: - development stages and development life cycle, system conception, domain analysis, application analysis, system design, class design, process summary

References:

1. Kendall & Kendall, “System Analysis & Design”, Prentice Hall India
2. Rumbaugh J., Blaha M., “Object-Oriented Modeling and Design with UML”, Pearson Education
3. Booch G., Rumbaugh J., Jacobson I., “The Unified Modeling Language - User Guide”, Pearson Education
4. Hoffer J., George J., et. al., “Modern System Analysis & Design”, Pearson Education

5. Fowler M., “UML Distilled”, Pearson Education
6. Bahrami A., “Object Oriented Systems Development using the Modified Modeling Language”, Tata McGraw Hill Publication
7. Brown D., “An Introduction to Object-Oriented Analysis – Objects and UML in Plain English”, John Wiley & Sons
8. Larman C., “Applying UML Patterns”, Pearson Education

Accomplishments of the student after completing the course:

After completion of the course the students would be well versed with

- The role of System Analyst
- Modern structured analysis approaches
- Key modeling concepts that apply to both the traditional structured approach and the newer object-oriented approach
- Unified Process and use of UML for Object-Oriented Analysis and Design

----

Course Name: Data Structures

Course Code: MCSCC222

Objectives:

The aim of this course is to enable students to

- Develop proficiency in the specification, representation, and implementation of data types and data structures
- Perform analysis of various algorithms for mainly time and space complexity
- Get a good understanding of applications of data structures
- Develop a base for advanced computer science study

Prerequisites:

Any programming language like C, C++

Contents:

1. **Analysis of algorithm and Datatypes**

   Data types, ADT, data structure: Definition & classification, Analysis of algorithms (recursive and non-recursive) with emphasis on best case, average case and worst case, Structures, Nesting of structures, Arrays of structures, Structures and Pointers, Structures and functions, Union and its usage, Pointer: Pointer arithmetic, Array and Pointers, Pointers and strings, Pointer to Pointer, Pointers and functions, Introduction to dynamic memory allocation, Allocating memory dynamically, resizing and releasing dynamically allocated memory

2. **Linear Data Structure with applications**

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

3. **Non-Linear data structures**

   Tree: Introduction and representation, Forest, Tree traversal, Binary Tree (representation using array and links), Binary tree, Traversal (recursive & non-recursive implementation), Expression tree, Graph: Introduction, Representations, Traversal (Breadth First Search, Depth First Search), Applications: Shortest path (Single source-all destinations), Minimal spanning tree (Prim's algorithm, Kruskal's algorithm)

4. **Searching and Sorting**

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

References:

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

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

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

----

Course Name: Web Application Development

Course Code: MCSCC223

Objectives:

The main objective of this course is to enable students to

- Learn the fundamentals of dynamic web applications
- Develop web applications using open source tools like PHP and MySQL
- Handle exceptions & Optimize websites using PHP

Contents:

1. **PHP Overview, File Handling, Arrays & String Manipulation**

   Overview of PHP:

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

   File Handling:

   Saving Data for Later, Processing Files, Opening a File, Writing to a File, Closing a File, Reading from File, Using other Useful File Functions, Locking Files

   Arrays:

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

   String Manipulation & Regular Expressions:

   Format Strings, Joining and Splitting Strings with String Functions, Compare Strings, Matching and Replacing Substrings with String Functions, Regular Expressions

2. **Reusing Code and Writing Functions, Object Oriented PHP, Exception Handling**

   Functions:

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

   Object-Oriented PHP:

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

   Error & Exception Handling:

   Exception Handling Concepts, The Exception Class, User-Defined Exceptions, Exceptions and PHP’s Other Error Handling Mechanism

3. **Designing Web Database**

   Relation Database Concepts, Designing Your Web Database, Web Database Architecture, Using the MySQL Monitor, Logging in to MySQL, Creating Databases and Users, Setting Up Users and Privileges, Introduction to MySQL’s Privilege System, Setting Up a user for the Web, Creating Database Tables, Understanding MySQL Identifiers, Choosing Column DataTypes

MySQL Database:

What is SQL, Inserting data into the Database, Retrieving Data from the Database, Updating Records in the Database, Altering Tables After Creation, Deleting Records from the Database, Dropping Tables, Dropping a Whole Database

Accessing MySQL Database using PHP:

How Web Database Architecture Work?, Querying a Database from the Web, Putting new information in the Database, Using Prepared Statements, Using Other PHP-Database Interfaces

4. Interacting with the File System and the Server, Date & Time, Session Control

Uploading Files, Using Directory Functions, Interacting with the File System, Using Program Execution Functions

Managing the Date & Time:

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

Using Session Control in PHP:

What is Session Control?, Understanding Basic Session Functionality, Implementing Simple Sessions, Creating a simple session, Configuring Session control, Implementing Authentication with Session Control

References:

1. Thomson L., Welling L., “PHP and MySQL Web Development”, Pearson Education
2. Gilmore W., “Beginning PHP and MySQL 5 from Novie to Professional”, Apress
3. Naramore E., Gerner J., et. al., “Beginning PHP5, Apache and MySQL Web Development”, Wrox Publication
4. Nixon R., “Learning PHP, MySQL and JavaScript”, O’Reilly Media

Accomplishments of the student after completing the course:

After completion of the course students will be able to

- Efficiently continue to expand their web development knowledge
- Build dynamic websites and web applications

Course Name: Data Structures - Practical

Course Code: MCSCC224

Objectives:

The main objective of this course is to enable students to

- Learn how to define structure, pointer
- Understand the concept of dynamic memory allocation
- Learn how to define, use and work with linear and non-linear data structure
- Learn about various searching and sorting techniques

Prerequisites:

Any programming language like C, C++

Contents:

1. **Structure, Pointer and Linear Data structures**

   Writing application programs to involving following data structures:
   - Structure, Union and their applications
   - Pointers and Dynamic Memory Allocation
   - Array and operations on array
   - Linked list(singly, doubly, circular), operations on linked list and its applications
   - Stack using array and linked list, operations on stack and its applications
   - Queue, operations on queue and its applications
   - Implementation of Hashing

2. **Non-Linear data structures, Searching and Sorting**

   Writing programs to implement the following:
   - Linear and Binary Search
   - Binary Search Tree, operations on binary search tree and its applications
   - Graph, operations on graph and its applications
   - Sorting with different algorithms
   - Implementation of Heap

Reference Book(s):

1. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, 2nd Edition, Pearson Education
2. G. A.V. PAI, “Data structures and algorithms, concepts, Techniques and Applications”, 1st Edition, TMH
3. Horowitz, Sahni, Anderson-Freed, “Fundamentals of Data Structures in C”, 2nd edition, University Press
4. Jean-Paul Tremblay, Paul G. Sorenson, “An Introduction to Data Structures with Applications”, 2nd Edition, Tata McGraw-Hill
5. Cormen, Leiserson, Rivest, Stein, “Introduction to Algorithm”, 2nd Edition, PHI

6. Gilberg & Forouzan, “Data Structures: A Pseudo-code Approach with C”, Thomson Learning
7. Parag Dave & Himanshu Dave, “Design and Analysis of Algorithms”, Pearson Education
8. Tanenbaum, “Data Structures Using C & C++” , PHI
9. Michel Goodrich, Roberto Tamassia, “Algorithm design-foundation, analysis & internet examples”, Wiley
10. A. V. Aho, J. E. Hopcroft, J. D. Ullman, “Data Structures & Algorithms”, Addison-Wesley Publishing
11. Michael Berman, “Data Structures Via C++: Objects by Evolution”, Oxford University Press
12. D. E. Knuth, “Sorting & Searching - The Art of Computer Programming”, Vol. 3, Addison-Wesley Publishing

Accomplishments of the student after completing the course:

After completion of the course students will be able to

- Develop programs with different linear and non-linear data structure
- Apply Searching and Sorting techniques using data structures

----

Course Name: Web Application Development - Practical

Course Code: MCSCC225

Objectives:

The main objective of this course is to enable students to

- Implement scripts of PHP
- Establish database connectivity to perform CRUD operations using PHP
- Build dynamic web applications

Contents:

1. **PHP Programming**
   Using variables, operators, functions, decision making, iterations
   Using files and performing file operations
   Using Array
   Performing String Manipulations & Using Regular Expressions
   Writing Functions & Reusing Code
   Using Object Oriented PHP
   Exception Handling
   Using inbuilt functions of PHP, Date & Time, etc.

2. **Using Databases**
   Creating Databases
   Applying basic principles of RDBMS
   Accessing Databases using PHP
   Manipulating Databases using PHP
   Using Session Control in PHP

References:

1. Thomson L., Welling L., “PHP and MySQL Web Development”, Pearson Education
2. Gilmore W., “Beginning PHP and MySQL 5 from Novie to Professional”, Apress
3. Naramore E., Gerner J., et. al., “Beginning PHP5, Apache and MySQL Web Development”, Wrox Publication
4. Nixon R., “Learning PHP, MySQL and JavaScript”, O’Reilly Media

Accomplishments of the student after completing the course:

After completion of the course students will be able to

- Efficiently continue to expand their web development knowledge
- Build dynamic websites and web applications

Course Name: Technical Writing & Communication Skills

Course Code: MCSAE221

Objectives:

The aim of this course is

- To acquaint the students with the basic concepts and techniques of communication that are useful in developing the skills of communicating effectively
- To develop reporting skills
- To inculcate technical & academic writing skills

Prerequisites: NIL

Contents:

1. **Technical Communication & Presentation - Strategies & Techniques**

Technical Communication: Features; Distinction between General and Technical Communication.

Technical Report: Definition & importance; Thesis/Project writing: structure & importance; synopsis writing: Methods; Technical research Paper writing: Methods & style; Seminar & Conference paper writing; Expert Technical Lecture: Theme clarity; Analysis & Findings; 7 Cs of effective business writing: concreteness, completeness, clarity, conciseness, courtesy, correctness, consideration, C.V./Resume writing; Technical Proposal: Types, Structure & Draft.

Writing as a part of Science, Characteristics of scientific writing, Thinking and Planning, choosing & Using words appropriately, Using numbers & illustrations, Use of phrases and clauses in sentences, importance of proper punctuation, creating coherence, organizing principles of paragraphs in documents, Nature and Style of Writing

Business Writing

Enquiries, Circulars, Quotations, Orders, Acknowledgments, Executions, Complaints, Claims and adjustments, Collection letter, Banking correspondence, Agency correspondence, Bad news and persuading letters, Sales letters, Job application letters – Biodata, Covering Letter, Interview Letters, Letter of Reference, Memos, minutes, Circulars & notices, Reports - Format, Choice of vocabulary, coherence and cohesion, paragraph writing, organization reports, by individual, Report by committee.

2. **Communication Skills**

Presentation: Forms; interpersonal Communication; Class room presentation; style; method; Individual conferencing: essentials: Public Speaking: method; Techniques: Clarity of substance; emotion; Humour; Modes of Presentation; Overcoming Stage Fear; Audience Analysis & retention of audience interest; Methods of Presentation: Interpersonal; Impersonal; Audience Participation: Quizzes & Interjections.

Interview skills; Group Discussion: Objective & Method; Seminar/Conferences Presentation skills: Focus; Content; Style; Argumentation skills: Devices: Analysis; Cohesion & Emphasis; Critical thinking; Nuances: Exposition narration & Description; effective business communication competence: Grammatical; Discourse competence: combination of expression & conclusion; Socio-linguistic competence: Strategic competence: Solution of communication problems with verbal and non-verbal means

Dimensions of Oral Communication & Voice Dynamics: Code and Content; Stimulus & Response; Encoding process; Decoding process; Pronunciation Etiquette; Syllables; Vowel sounds; Consonant sounds; Tone: Rising tone; Falling Tone; Flow in Speaking; Speaking with a purpose; Speech & personality; Professional Personality Attributes: Empathy; Considerateness; Leadership; Competence

References:

1. Raman M. & Sharma S., "Technical Communication – Principles and Practices", Oxford University Press
2. Mitra B., "Personality Development", Oxford University Press
3. Bansal R., Harrison J., "Spoken English – A Manual of Speech Phonetics", Orient Blackswan, New Delhi
4. Sharma R., Mohan K., "Business Correspondence and Report Writing", Tata McGraw Hill & Co., New Delhi
5. Pandey L, "Practical Communication: Process and Practice", AITBS Publications India Ltd., Delhi
6. Theodore S., "Modern Technical Writing", Prentice Hall
7. Sharma S.D., "A Text Book of Scientific & Technical Writing", Vikas Publication, Delhi
8. Murphy M., "Skills for Effective Business Communication", Harward University
9. Mehra P., "Business Communication for Managers", Pearson Publication, Delhi

Accomplishments of the student after completing the course:

After completion of this course, students will be able to

- Understand the nature and objective of Technical Communication relevant for the work place as Engineers
- Utilize the technical writing for the purposes of Technical Communication and its exposure in various dimensions
- Imbibe inputs by presentation skills to enhance confidence in face of diverse audience
- Evaluate their efficacy as fluent & efficient communicators by learning the voice-dynamics

----

Course Name: Project - 2

Course Code: MCSSE221

Objectives:

The aim of this course is

- To introduce structured development of software systems
- To acquaint students to various techniques of requirements determination
- To introduce the concepts of analysis and design
- To model the system with various software diagrams
- To develop a system using software engineering concepts
- To prepare document/report of the system

Prerequisites:

Knowledge of Programming

Contents:

1. **Introduction to System Analysis & Design**
   Concepts of System Analysis & Design, Components of a System, Users of a System, System Development Life Cycle, Role of System Analyst, Various approaches to develop a system, Implementing Security in Systems

2. **Structured Systems Development Approach**
   Requirements Gathering, surveying existing systems, Performing Feasibility Study before building a system, Formulating SRS, Designing Data Flow Diagrams, Entity-Relationship Diagrams, Designing Data Dictionary, Designing effective Input and Reports, Building Test Cases, Documenting the system.

Guidelines for Project:

1. Projects can be created by a single student or in a group of 2-3 students.
2. The definition of project is to be submitted within 15 days of the starting of the semester.
3. The project should be free from plagiarism.
4. A student/group is required to report to their project guides on regular basis.
5. A student/group is required to present the project during internal exams.
6. It is advisable that the project must include different forms and reports.
7. It is necessary to prepare system diagrams, data dictionary as per the convention.
8. Project documentation should be prepared for the semester end examination.
9. In the semester end examination, the project should be presented to the examiners along with the documentation(report).

References:

1. Kendall & Kendall. “System Analysis and Design ”, Prentice Hall India (PHI)
2. James Sen, “Analysis and Design of Information System”, PHI

3. Arthur Langer, “Analysis and Design of Information Systems “, Springer
4. Jeff Johnson, “Designing with the Mind in Mind “, Morgan Kaufmann
5. Dennis, Wixom, Roth “System Analysis and Design”, Wiley India
6. Joseph Valacich, Joey F. George, “Modern Systems Analysis and Design”, Pearson
7. Priti Srinivas Sajja., “Essence of System Analysis and Design”, Springer

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Analyse and Design a system based on user requirements
- To develop a software system using software engineering concepts
- To prepare documention for the system

Course Name: Programming with Python

Course Code: MCSGE221

Objectives:

The main objective of this course is to enable students to

- Understand the core concepts of python programming
- Understand basic insight of programming using python functions
- Give hands on with major focus on practical implementation of object-oriented concepts

Prerequisites:

Fundamentals of Computers

Contents:

1. **Introduction to Python**

   Features of Python, Comparison between C and Python, Data types in Python, Type of Comments, Built-in data type, Bool data type, Sequences in Python, Sets, Literals in python, Operators, Identity Operators and Membership Operators in Python, Arrays in Python, Control structure – Condition execution in Python, Break Statement, Continue Statement, Pass Statement, Assert and Return Statement, Using iteration within Python programs, Arrays in Python

   Lists: Using Lists, List Traversal, Building Lists, List Membership, List Assignment and Equivalence, List Bounds, Slicing, List Element Removal Lists and Functions, List Methods, List Comprehensions, Multidimensional Lists, Lists Vs. Generators

   Tuples, Dictionaries, Sets: Tuples, Arbitrary Argument Lists, Dictionaries, Using Dictionaries, Counting with Dictionaries, Grouping with Dictionaries, Keyword Arguments, Sets, Set Quantification with all and any, enumerating the Elements of a Data Structure

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

2. **Functions and Object-Oriented Programming in Python**

   Function: Difference between a Function and a Method, Defining a Function, Calling a Functions, Returning Results from a Function, Returning Multiple Values from a Function, Functions are First Class Objects, Pass by Object reference, Formal and Actual Arguments, Positional Arguments, Keyword Arguments, Default Arguments, Variable Length Arguments, Local and Global Variables, The Global Keyword, Passing a Group of Elements to a function, Recursive Functions, Anonymous Functions or Lambdas, Function Decorators, Generators, Structured Programming, Creating our Own Modules in Python

   Object Oriented Programming: Introduction to Object Oriented Programming, Classes and Objects, Inheritance in Python Class, Encapsulation and Polymorphism, Abstract Classes and Interfaces, Exception handling

References:

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

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Gain awareness about various python programming core concepts, modularity concept and object-oriented concepts
- Solve challenging problems using python programming language

----

5 Years Integrated MSc (Computer Science)

[Course Initiated in 2021-2023]

Semester - V

Scheme & Course Structure

Applicable from July, 2023

Course Structure with Credits

	Course Type	Subject Names & Credits	Total Credits
Semester-V	Core Courses	• MCSCC311-Operating Systems (4) • MCSCC312-Java Programming (4) • MCSCC313- Advanced Networking (4) • MCSCC314- Java Programming – Practical (2) • MCSCC315- Advanced Networking – Practical (2)	16
	Ability Enhancement Courses	• MCSAE311- Critical Thinking (2)	2
	Skill Enhancement Courses	• MCSSE311- Project – 3 (4)	4
	Discipline Specific Elective Courses		—
	Generic Elective Courses	• MCSGE311 – Cyber Security (2)	2
	Semester-V (Total Credits)		24

TEACHING & EXAMINATION SCHEME FOR 5 YEAR INTEGRATED M. Sc. (COMPUTER SCIENCE) COURSE

Semester-V	Course		Hours per week								Credits
	Course Code	Name	Theory	Practical	Internal	Total	Lectures	Others (Tutorials)	Practical	Total	
Semester-V	MCSCC311	Operating Systems	70	***	30	100	3	1	***	4	4
	MCSCC312	Java Programming	70	***	30	100	3	1	***	4	4
	MCSCC313	Advanced Networking	70	***	30	100	4	***	***	4	4
	MCSCC314	Java Programming – Practical	***	35	15	50	***	***	4	4	2
	MCSCC315	Advanced Networking – Practical	***	35	15	50	***	***	4	4	2
	MCSSE311	Project – 3	***	70	30	100	***	***	8	8	4
	MCSAE311	Critical Thinking	35	***	15	50	2	***	***	2	2
	MCSGE311	Cyber Security	35	***	15	50	2	***	***	2	2
		Total	280	140	180	600	14	2	16	32	24

Theory/Tutorial Credits → 1 Hour = 1 Credit

Practical Credits → 2 Hours = 1 Credit

Course Name: Operating Systems

Course Code: MCSCC311

Objectives:

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

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

Prerequisites:

Fundamentals of Computers

Contents:

1. **Overview of Operating Systems**

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

2. **Process Management & Synchronization**

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

   Process Synchronization:

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

3. **Memory Management & Storage Management**

   Memory Management:

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

   Storage Management:

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

4 File System & Basics of Linux

File System:

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

Case Study of Linux OS:

Basic commands of Linux, Process related commands of Linux, Linux APIs for process management and Inter Process Communication, Linux Memory Management, File Systems

References:

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

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Gain working knowledge of Linux System
- Harness the facilities provided by Operating System in application development
- Optimize the programs for specific Operating System

Course Name: Java Programming

Course Code: MCSCC312

Objectives:

The main objective of this course is to enable students to

- Understand the concepts of Object Oriented Programming Language and easily use Java
- Get good understandings of developing platform independent applications using the Java Programming Language

Prerequisites:

Basic knowledge of C/C++ programming language

Contents:

1. **Introduction, Data Types, Variables and Constants, Loops and Logic**
   Features of the Java Language, Java Environment, Object Oriented Programming in Java, Java Program Structure, Java and Unicode, Data and Variables, Integer Data Types, Floating Point Data Types, Arithmetic Calculations, Mixed Arithmetic expressions, Mathematical functions and constants, Bitwise operators, Enumerated data type, Boolean Variables Operator precedence, Program comments, Loops and Logic Making decisions, Logical operators, The Conditional Operator, The switch Statement, Variable scope, Loops, Assertions.

2. **Arrays, Strings, Classes, Extending Classes & Inheritance**
   Arrays, Strings, Operations on Strings, Mutable Strings, Significance of class, Defining classes, Defining methods, Defining constructors, Defining and using a class, Method overloading, Using Objects, Understanding packages, controlling access to class members, Nested classes, The finalize() method, Native method, Class inheritance, choosing base class access attributes, Polymorphism, Multiple Levels of inheritance, Abstract classes, The Universal superclass, Methods accepting a variable number of arguments, Casting, Object, Enumeration in detail, Using the final modifier, Interfaces, Anonymous classes

3 **Exceptions, File Handling & I/O Operations, Threads**
   Types of Exceptions, Dealing with exceptions, Exception Objects, Defining your own exceptions, The File Class, I/O Stream, Classes for Reading & Writing to Files, RandomAccessFile Class, Serializing Objects
   Understanding Threads, the Thread class, creating thread of executing using Runnable and using sub-class of Thread, various properties of Thread, using synchronized keyword, using wait and notify

4 **Collection Framework, Few other useful classes**
   The Collection Framework from java.util package, Collection, Iterable, List, Set, Map, Queue, Dequeue, utility methods from Collections class, java.time package, classes, LocalDate, LocalTime, LocalDateTime, OffsetDate, ZonedDateTime, Duration, Period, Clock and Instant classes, from java.util package, understanding the Optional classUtility Methods for Arrays, Generating Random Numbers, Dates and Times, Regular Expressions, Date and Time APIs, Lambda Expressions, Type Annotations, Method References, Collectors Class

References:

1. Cay S Horstmann, Gary Cornell, “Core Java, Volume 1 – Fundamentals”, Pearson Education
2. Jain Pravin, “The class of Java”, Pearson India

3. Horton I., "Beginning Java", 7th ed., Wrox India
4. Ken Arnold, James Gosling, David Holmes, "The Java Programming Language", Addison-Wesley Pearson Education.
5. Subramian V., "Functional Programming in Java", Pragmatic Bookshelf
6. Urma R., Fusco M., Mycroft A., "Java 8 in action", Manning Publications

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Create appropriate classes using the Java Programming Language to solve a problem using Object Oriented Approach and use concepts of Functional programming
- Develop platform independent applications using the Java Programming Language

----

Course Name: Advanced Networking

Course Code: MCSCC313

Objectives:

The objective of the course is to enable students to

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

Prerequisites:

Linux Operating Systems

Contents:

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

2. **The Application Layer & Transport Layer**
   Application Layer:
   Principles of Network Applications, The Web and HTTP, E-mail Services and its Protocols, DNS, FTP, SSH and Telnet, Peer-to-Peer Applications, Video Streaming & Content Distribution Networks, Overview of Socket Programming for designing Network based applications
   Transport Layer:
   Transport Layer Services, Multiplexing and Demultiplexing, Connectionless Transport, Principles of Reliable Data Transfer, Connection Oriented Transfer, Principles of Congestion Control, TCP Congestion Control, Variants of TCP

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

4. **Link Layer, LANs & Wireless Networks**
Link Layer:
Introduction, Error-Detection and Error-Correction Techniques, Multiple Access Links & Protocols, Switched LANs, Link Virtualization
Wireless Networks:
Introduction, Wireless Links & Network Characteristics, WiFi 802.11 and Wireless LANs, Overview of Sensor Networks & IoT

References:

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

Accomplishments of the student after completing the course:

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

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

----

Course Name: Java Programming - Practical

Course Code: MCSCC314

Objectives:

The aim of this course is

- Understand the concepts of Object Oriented Programming Language and easily use Java
- Get good understandings of developing platform independent applications using the Java Programming Language

Prerequisites:

Basic knowledge of C/C++ programming language

Contents:

1. **Basics of Java Programming**
   Using Integrated Development Environment for Java Programming, Compiling & Executing Java Programs, Using Different Data Types in Java, Using various Operators & Performing Arithmetic Calculations, Decision Making with Java Programming, Looping, Implementation of Assertions, Use of Classes in Java, Constructors in Java, Exception Handling

2. **Object Oriented Java Programming**
   Implementation of Inheritance using Classes & Interfaces, Implementing Lambda, Implementation & Use of Packages in Java, Commonly used classes & Methods, File Handling using different techniques, Implementing Thread Class & Runnable Interface in Java, Implementation of various collection frameworks

References:

1. Cay S Horstmann, Gary Cornell, “Core Java, Volume 1 – Fundamentals”, Pearson Education
2. Jain Pravin, “The class of Java”, Pearson India
3. Horton I., “Beginning Java”, 7th ed., Wrox India
4. Ken Arnold, James Gosling, David Holmes, “The Java Programming Language”, Addison-Wesley Pearson Education.
5. Subramian V., “Functional Programming in Java”, Pragmatic Bookshelf
6. Urma R., Fusco M., Mycroft A., “Java 8 in action”, Manning Publications

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Create appropriate classes using the Java Programming Language to solve a problem using Object Oriented Approach and use concepts of Functional programming
- Develop platform independent applications using the Java Programming Language

----

Course Name: Advanced Networking - Practical

Course Code: MCSCC315

Objectives:

The aim of this course is

- Implement Client Server based programs using Network Programming
- Understand the implementation of networking related functionalities provided by programming language

Prerequisites:

Basic knowledge of Java / Python programming language

Contents:

1. **Basics of Network Programming**
   Overview of Client-Server Networking, Functions provided by programming language for Network Programming, Using IP Address & Port Numbers in Network Programming, Implementing IP Address Scanner & Port Scanner, Fetching Network Configuration Details using Programming Language, Use of URLs, DNS in Networking, Studying different exceptions in Network Programming, Use of tools like nmap and Wireshark

2. **Building Networking Based Applications**
   Socket programming using functionalities of User Data Protocol, Socket programming using functionalities of Transmission Control Protocol, Building basic Client-Server applications like Data-Time Services, File Services, DNS Services, DHCP Services, etc.

References:

1. Stevens, Comer D., Douglas E., “Internetworking with TCP/IP Vol. III”, Prentice Hall
2. Galbraith J., “Network Programming in Python: The Basic: A Detailed Guide to Python 3 Network Programming and Management”, BPB Publication
3. Neidinger M., “Python Network Programming Techniques”, Packt Publishing Ltd.
4. Harold E.R., “Java Network Programming”, O’Reilly Media

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Build simple networking utilities
- Build Client-Server applications

Course Name: Critical Thinking

Course Code: MCSAE311

Objectives:

The main objective of this course is to enable students to

- Rigorously understand and evaluate arguments and evidence, and to use these things to come to a reasoned judgment
- Enhance the ability to become a good critical thinker

Prerequisites:

NIL

Contents:

1. **Arguments & Generalizations**

   Overview of Critical Thinking, Role of Reasoning & Arguments

   Short Arguments: Identifying premises and conclusion, Developing Ideas in a natural order, Starting from reliable premises, Being concrete and concise, Building on substance, Use of consistent terms

   Generalizations: Using multiple examples, Using representative examples, Identifying relative background rates, Working with statistics, Considering counter examples, Evaluating arguments for Generalizations

   Arguments by Analogy: Identifying important similarities & differences, Evaluating arguments by analogy, Constructing arguments by analogy

   Sources: Seeking information from sources, Citing sources, Identifying biased sources, Identifying independent sources, Evaluating arguments that use sources, Using sources in arguments

2. **Arguments & its Types**

   Arguments about Causes: Causal Arguments, Correlations, Most likely explaniation in arguments, Complexity, Evaluating arguments about causes, Constructing arguments about causes

   Deductive Arguments: Modus ponens, Modus tollens, Hypothetical syllogism, Disjunctive syllogism, Identifying deductive arguments, Drawing conclusions, Reductio ad absurdum, Deductive arguments in several steps

   Extended Arguments: Exploring the issue, Spelling out basic ideas as arguments, Defending basic premises, Considering objections, Considering alternatives

   Oral Arguments: Reaching out to the audience, Signposting arguments, Reframing arguments, Ending in Style, Evaluating Oral Presentations

   Argumentative Essays: Writing good leads, Making a definite claim or proposal, Writing out the arguments, Detailing and Meeting objections

References:

1. Morrow D., Weston A., "A Workbook for Arguments – A Complete Course in Critical Thinking", 2nd Edition, Hackett Publishing Company Inc.
2. Haber J., "Critical Thinking", MIT Press
3. Inch E., Warnick B., "Critical Thinking & Communication", Pearson India
4. Fisher A., "Critical Thinking – An Introduction", Cambridge University Press

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Effectively apply reasoning on different situations
- Think Critically on various scenarios
- Effectively communicate with the team and clients of IT Industry

----

Course Name: Project - 3

Course Code: MCSSE311

Objectives:

The aim of this course is

- To introduce structured development of software systems
- To acquaint students to various techniques of requirements determination
- To introduce the concepts of analysis and design
- To model the system with various software diagrams
- To develop a system using software engineering concepts
- To prepare document/report of the system

Prerequisites:

Knowledge of Programming

Contents:

1. **Introduction to System Analysis & Design**
   Concepts of System Analysis & Design, Components of a System, Users of a System, System Development Life Cycle, Role of System Analyst, Various approaches to develop a system, Using a "systems" approach to build whole system

2. **Structured Systems Development Approach**
   Requirements Gathering, surveying existing systems, Performing Feasibility Study before building a system, Formulating SRS, Designing Data Flow Diagrams, Entity-Relationship Diagrams, Designing Data Dictionary, Designing effective Input and Reports, Implementing EOD, EOM and EOY Procedures, Documenting the system

Guidelines for Project:

1. Projects can be created by a single student or in a group of 2-3 students.
2. The definition of project is to be submitted within 15 days of the starting of the semester.
3. The project should be free from plagiarism.
4. A student/group is required to report to their project guides on regular basis.
5. A student/group is required to present the project during internal exams.
6. It is advisable that the project must include different forms and reports.
7. It is necessary to prepare system diagrams, data dictionary as per the convention.
8. Project documentation should be prepared for the semester end examination.
9. In the semester end examination, the project should be presented to the examiners along with the documentation (report).

References:

1. Kendall & Kendall. "System Analysis and Design ", Prentice Hall India (PHI)
2. James Sen, "Analysis and Design of Information System", PHI
3. Arthur Langer, "Analysis and Design of Information Systems ", Springer

4. Jeff Johnson, “Designing with the Mind in Mind “, Morgan Kaufmann
5. Dennis, Wixom, Roth “System Analysis and Design”, Wiley India
6. Joseph Valacich, Joey F. George, “Modern Systems Analysis and Design”, Pearson
7. Priti Srinivas Sajja., “Essence of System Analysis and Design”, Springer

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Analyse and Design a system based on user requirements
- Develop a software system using software engineering concepts
- Prepare a document for the system

Course Name: Cyber Security

Course Code: MCSGE311

Objectives:

The main objective of this course is to enable students to

- Understand the cyber security threat landscape
- Develop a deeper understanding and familiarity with various types of cyberattacks, cybercrimes, vulnerabilities and remedies thereto
- Analyze and evaluate existing legal framework and laws on cyber security
- Analyze and evaluate the digital payment system security and remedial measures against digital payment frauds
- Take measures for self-cyber-protection as well as societal cyber-protection

Prerequisite: Nil

Contents:

1. **Cyber Security & Cyber Crimes, Vulnerabilities & Management Practices**

   Introduction to Information Systems, Evolution and Types of Information Systems, Evolution and Growth of the Internet, Information System Development and Maintenance, Cyberspace and Cybersecurity, Overview of Information Security, Organization and Governance of the Internet and Cybersecurity, Information Security Models

   Cybercrimes, Classification of Cybercrimes, Forms of Cybercrimes, Cyber Scams and Frauds, Sources of Threats: Threat Actors and their Motivations, Tools and Methods Used in Cyberattacks/Cybercrimes, Overview of a Cyberattack, Responding to Cyberattacks and the Cyber Kill Chain, Cyberattacks: Organizational Implications, Cyberattacks Impacting Citizens and Communities, Prevention of Cybercrime, International Efforts to Deal with Cybercrime, National Cybersecurity Policy, Online Code of Conduct and Computer Ethics

   Introduction, Security Considerations and Challenges, Types of Vulnerabilities, Project OWASP, Vulnerabilities Assessment, Common Vulnerabilities and Exposures (CVE): Institutional Mechanisms

   Overview of Cybersecurity Management, Information Classification Process, Security Policies, Security Procedures and Guidelines, Security Controls, Security Organization, Incident Response, Business Continuity and Disaster Recovery

2. **Cyber Law, Cybersecurity in E-commerce**

   Information Security Governance and Risk Management, Cyber Risk Management, Cybersecurity Frameworks, Cyber Resilience, Industry-specific Cybersecurity Frameworks, The Human Factor in Cybersecurity, Algorithms and Techniques for Cybersecurity

   Introduction, Securing Networks, Web Applications, Services and Servers, Email Security, Antivirus Technologies and Solutions, Identity and Access Management, Authentication, Cryptography, How Do Digital Money, Cryptocurrency and NFTs, Digital Signatures, Advanced Technologies and Approaches in Cybersecurity, Internet Protocols and Ports Need for Cyber Laws and Regulations, Role of International Law and Governments, Challenges for Law-makers and Law Enforcement Agencies, Cybersecurity Regulations, Cyber Forensics, Cybercrime Techniques, Prevention of Cybercrime and Protection, Cybercrime Investigation, Evidence Collection and Analysis, Intellectual Property Issues in Cyberspace

Introduction to Personal Cybersecurity, Common Causes of Personal Security Breaches, Personal Cybersecurity Best Practices, Privacy Regulations and Cybersecurity, Ethics in CyberSecurity

Future Challenges in Cybersecurity, Web 3.0, Harnessing Artificial Intelligence for Cybersecurity, Blockchain for Cybersecurity, Quantum Computing and Cybersecurity, Combating Advanced Persistent Threats, Digital Trust and Identity Management, 5G Networks and Cybersecurity, Adopting a 'Secure-by-Design' Approach, Supply Chain Cybersecurity

Introduction to E-commerce, Elements of E-commerce Security, E-commerce Security Best Practices, Digital Payments, RBI Guidelines on Digital Payments and Customer Protection, Laws on Privacy and Data Protection for E-commerce Companies

Introduction to Social Networks, Types of Social Media and Popular Platforms, Social Media Marketing, Social Media Monitoring, Social Media Privacy, Social Media Privacy Laws and Personal Data Protection, Flagging and Reporting of Inappropriate Content, Laws Regarding Posting of Inappropriate Content, Data Harvesting and Personal Data Protection, Best Practices for the Use of Social Media

References:

1. Singh Ajay, “Introduction to Cyber Security”, University Press
2. Denning D., “Information Warfare & Security”, Addison Wesley
3. Belapure S., Godbole N., “Cyber Security Understanding Cyber Crimes, Computer Forensics & Legal Perspectives”, Wiley India
4. Oliver H., “Security in the Digital Age: Social Media Security Threats and Vulnerabilities”, Create Space Independent Publishing Platform
5. Venkartarman N. & Shriram A., “Data Privacy Principles and Practic”, CRC Press
6. Kragbrothy W., “Information Security Governance, Guidance for Information Security Managers”, Wiley Publication
7. Weiss M., Solomon M., “Auditing IT Infrastructure for Compliance”, Jones Bartlett Learning

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Strengthen the cyber security of end user, IT infrastructure
- Effectively report cyber threats and cyber-crimes through the prescribed legal and government channels
- Perform risk based assessment, cyber security audit & compliance

----

5 Years Integrated MSc (Computer Science)
[Course Initiated in 2021-2023]

Semester - VI

Scheme & Course Structure

Applicable from July 2023

Course Structure with Credits

	Course Type	Subject Names & Credits	Total Credits
Semester-VI	Core Courses	• MCSCC321-Software Engineering (4) • MCSCC322-Distributed Application Development (4) • MCSCC323-Distributed Application Development - Practical (2)	**10**
	Ability Enhancement Courses	• MCSAE321- Management Information Systems	**2**
	Skill Enhancement Courses	• MCSSE321-Project-4 (4)	**4**
	Discipline Specific Elective Courses	• MCSDE321-Elective-1 (4) • MCSDE322-Elective-1 – Practical (2)	**6**
	Generic Elective Courses	• MCSGE321 – Mobile Application Development (2)	**2**
	Semester-VI (Total Credits)		**24**

TEACHING & EXAMINATION SCHEME FOR 5 YEAR INTEGRATED M. Sc. (COMPUTER SCIENCE) COURSE

Semester-VI	Course		Hours per week								Credits
	Course Code	Name	Theory	Practical	Internal	Total	Lectures	Others (Tutorials)	Practical	Total	
Semester-VI	MCSCC321	Software Engineering	70	***	30	100	3	1	***	4	4
	MCSCC322	Distributed Application Development	70	***	30	100	3	1	***	4	4
	MCSCC323	Distributed Application Development - Practical	***	35	15	50	***	***	4	4	2
	MCSAE321	Management Information Systems	35	***	15	50	2	***	***	2	2
	MCSSE321	Project – 4	***	70	30	100	***	***	8	8	4
	MCSDE321	Elective-1	70	***	30	100	3	1	***	4	4
	MCSDE322	Elective-1 – Practical	***	35	15	50	***	***	4	4	2
	MCSGE321	Mobile Application Development	***	35	15	50	***	***	4	4	2
		Total	245	175	180	600	11	3	20	34	24

Theory/Tutorial Credits → 1 Hour = 1 Credit

Practical Credits → 2 Hours = 1 Credit

Track wise subjects for Elective-1			
Track	**Web Technologies**	**Artificial Intelligence & Machine Learning**	**Security**
Subject Code	MCSDE321(1)	MCSDE321(2)	MCSDE321(3)
Subject Name	Web Framework	Artificial Intelligence	Foundations of Cryptography
Subject Code	MCSDE322(1)	MCSDE322(2)	MCSDE322(3)
Subject Name	Web Framework - Practical	Artificial Intelligence – Practical	Foundations of Cryptography - Practical

Course Name: Software Engineering

Course Code: MCSCC321

Objectives:

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

- Understand the basic concept and importance of Software Engineering
- Understand the process of Software Engineering
- Understand all the activities required to develop and maintain software

Prerequisites:

Basic knowledge of Programming, Systems Analysis and Design

Contents:

1. **Software Engineering, Process Models, Requirements Engineering & Modeling**
   Nature of Software, Unique nature of WebApps, Software Engineering, Software Engineering Practice and Principles, Generic Process Model, Process Assessment and Improvement, Prescriptive and Specialized Process Models, Unified Process, Personal and Team Process Model, Process Technology, Product and Process, Agility, Agility and cost of change, Agile Process
   Software Engineering Knowledge, Core Principles, Framework Activity Guiding Principles, Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Developing Use Cases, Building the Requirements Model, Negotiating Requirements, Validating Requirements, Requirement modeling Strategies, Flow-Oriented Modeling, Behavioral Model, Patterns of Requirements Modeling, Requirement Modeling of WebApps

2. **Software Design –Concepts, Architecture, Component, User Interface, Web**
   Design within the Context of Software Engineering, The Design Process, Design Concepts, The Design Model, Software Architecture, Architectural Genres, Architectural Styles, Architectural Design, Introduction of Component, Designing class-based Components, Conducting Component level Design, Component level design for WebApps, Designing Traditional Components, Component based Development, Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, WebApp Interface Design, Design Evolution, WebApp Design Quality, Design Goals, Design Pyramid for WebApps, WebApp Interface Design, Aesthetic Design, Content Design, Architecture Design, Navigation Design, Component level Design, Object oriented Hypermedia Design Method

3. **Quality Concepts, Review Techniques and Software Quality Assurance**
   Introduction of Quality, Software Quality, Software Quality Dilemma, Achieving Software Quality, Cost Impact of Software Defects, Defect Amplification and Removal, Review Metrics and its Use, Reviews: A Formality Spectrum, Informal Review, Formal Technical Reviews, Background Issues of SQA, Elements of Software Quality Assurance, SQA Tasks, Goals and Metrics, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, ISO - 9000 Quality Standards, SQA Plan

4. Software Testing, Software Configuration Management & Software Metrics

A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, Test for Object Oriented Software, Test Strategies for WebApps, Validation Testing, System Testing, Art of Debugging, Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, Basis Path Testing, Control Structure Testing, Black-Box Testing, Model Based Testing, Testing for Specialized Environments, Architecture, and Applications, Patterns of Software Testing, Broadening the View of Testing, Testing OOA and OOD Models, Object oriented Testing Strategies, Object oriented Testing Methods, Testing Methods Applicable at Class Level, Interclass Test-Case Design, Testing Concepts for WebApps, Overview of Testing Process, Content Testing, User Interface Testing, Component-level Testing, Navigation Testing, Configuration Testing, Security Testing, Performance Testing, Software Configuration Management, SCM Repository, SCM Process, Configuration Management for WebApps, Framework for Product Metrics, Metrics for Requirements Model, Metrics for the Design Model, Design Metrics for WebApps, Metrics for Source Code, Metrics for Testing, Metrics for Maintenance

References:

1. Roger Pressman, “Software Engineering – A Practitioner’s Approach”, 7th Edition, McGraw Hill Higher Education
2. Sommerville, “Software Engineering”, Pearson Education
3. W S Jawadekar, “Software Engineering – Principles and Practices”, TMH Publication
4. S A Kelar, “Software Engineering – A Concise Study”, PHI Publication

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Understand and appreciate the importance of Software Engineering in today’s world
- Understand and perform the various activities required to develop good quality software within time and cost budget

----

Course Name: Distributed Application Development

Course Code: MCSCC322

Objectives:

The main objective of this course is to enable students to

- Apply the concepts of data sharing across heterogeneous systems
- Develop Client Server applications using Java Servlets
- Develop Client Server applications using Java Server Pages

Prerequisites:

HTML, DHTML and Java Programming

Contents:

1. **XML, JSON Technologies & Generics in Java**

   Introducing XML, Parsing an XML Document, Validating XML Documents, Locating information with XPath, Using Namespaces, Streaming Parser, Generating XML Documents, XSL Transformations, Overview of JSON, Consuming JSON data using programming languages, Overview of Web Services

   Generic Class Type:

   What are Generic types?, Defining a Generic Class Type, Generic Type and Generic Interfaces, Variables of a Raw Type, Using wildcards as type parameter arguments, Arrays and parameterized types

2. **Web Application Components**

   Understanding web applications, Introducing the MVC design pattern, JAR file, WAR file, HTTP, GET request method, POST request method, GET and POST in HTML processing, other Request methods, The HTTP Response, Using Deployment Descriptors

3 **Servlets**

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

4 **Java Server Pages**

Overview of JSP Technology, Invoking Java Code with JSP Scripting elements, controlling the structure of generated servlets: The JSP Page directive, including files and Applets in JSP Pages, Using JavaBeans Components in JSP Documents, Integrating Servlets and JSP: The MVC Architecture, Simplifying access to Java Code: The JSP 2.0 Expression Language, Using JSTL, Tag Libraries: Basics, JSP Standard Tag Library

References:

1. Ivan Bayross, “Web Enabled Commercial Application Development Using HTML,DHTML, PERL, Java Script”, BPB publications, Revised Edition
2. Dave Crane , Bear Bibeault and Tom Locke, “Prototype and Scriptaculous in Action”, Manning Publication
3. Marty Hall and Larry Brown, “Core Servlets and JavaServer Pages Volume - 1”, Pearson Education, 2nd Edition
4. Marty Hall and Larry Brown, “Core Servlets and JavaServer Pages Volume - 2”, Pearson Education, 2nd Edition
5. Chuck Cavaness , “Programming Jakarta Struts”, O’Reilly Publication
6. Sue Spielman and Meeraj Kunnumpurath , “Pro J2EE 1.4 From Professional to Expert” , APress Publication
7. Robert W. Sebsta, “Programming the World Wide Web”, Pearson Education

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Develop platform independent web applications
- Apply web technologies concepts to develop distributed web applications

----

Course Name: Distributed Application Development - Practical

Course Code: MCSCC323

Objectives:

The objective of the course is to enable students to

- Apply the concepts of data sharing across heterogeneous systems
- Develop Client Server applications using Java Servlets
- Develop Client Server applications using Java Server Pages

Prerequisites:

HTML, DHTML and Java Programming

Contents:

1. **XML, JSON Technologies & Generics in Java**
   Creating XML Documents, Parsing XML Documents, Validating XML Documents, locating information with XPATH, Applying XSL Transformations, Using Namespaces, Streaming Parser, Generating JSON Documents, Parsing JSON Documents, Creating & Consuming Web Services, Applying Generic class and interfaces in Java programming, Applying wildcards, Implementing arrays and parameterized types

2. **Web Application Development**
   Dealing with JAR file & WAR files, HTTP, Processing of GET and POST in HTML, The HTTP Response, Using Deployment Descriptors, Implementing client server applications using Java Servlets, Implementing client server applications using Java Server Pages, Using JSTL, Developing applications using MVC architecture

References:

1. Ivan Bayross, “*Web Enabled Commercial Application Development Using HTML,DHTML, PERL, Java Script*”, BPB publications, Revised Edition
2. Dave Crane , Bear Bibeault and Tom Locke, “Prototype and Scriptaculous in Action”, Manning Publication
3. Marty Hall and Larry Brown, “Core Servlets and JavaServer Pages Volume - 1”, Pearson Education, 2nd Edition
4. Marty Hall and Larry Brown, “Core Servlets and JavaServer Pages Volume - 2”, Pearson Education, 2nd Edition
5. Chuck Cavaness , “Programming Jakarta Struts”, O’Reilly Publication
6. Sue Spielman and Meeraj Kunnumpurath , “Pro J2EE 1.4 From Professional to Expert” , APress Publication
7. Robert W. Sebsta, “Programming the World Wide Web”, Pearson Education

Accomplishments of the student after completing the course:

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

- Develop platform independent web applications
- Apply web technologies concepts to develop distributed web applications

----

Course Name: Management Information Systems

Course Code: MCSAE321

Objectives:

The main objective of this course is to enable students to

- Establish the role of information systems in organizations and how they relate to organizational objectives and organization structure
- Provide knowledge of basic techniques and elementary skills in representing system structure
- Introduce various types of applications which are part of an information system

Prerequisites:

Basic knowledge of C/C++ programming language

Contents:

1. **Introduction to Systems**
   The Organizational Structure, Business Process, Business Considerations – Accuracy, Queues, Security, Priorities, Audit-Trails, Report Frequency, Exception Reporting, Documentation, User Acceptance, EOD/EOM/EOY Procedures, ERP Systems, Components of an ERP System, Data in ERP Systems, Reporting in ERP, Case Study of an ERP

2. **Functions and Object-Oriented Programming in Python**
   Accounting Systems, Procurement Process & Systems, Fulfilment Process, Production Process, Inventory & Warehouse Management Process, Material Planning Process, Process Integration, Systems Design: Study of Customer Order Processing & Invoicing System, Inventory Management Systems – Stock Replenishment & Inventory Management, Accounts Payable System, Accounts Receivable System, Financial Accounting System, Payroll System, System Maintenance

References:

1. Magal S., Word J., “Integrated Business Processes with ERP Systems”, John Wiley & Sons Inc.
2. Haueisen W., Camp J., “Business Systems for Microcomputers – Concept, Design & Implementation”, Prentice Hall India Pvt. Ltd.
3. Murdick R., Ross J., Clagget J., “Information Systems for Modern Management”, Prentice Hall India Pvt. Ltd.
4. Jawadekar W., “Management Information Systems – Text & Cases”, McGraw Hill

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Gain awareness about various components of ERP Systems

- Build solutions for Management Information Systems, Decision Support Systems

----

Course Name: Project – 4

Course Code: MCSSE321

Objectives:

The aim of this course is

- To introduce structured development of software systems
- To acquaint students to various techniques of requirements determination
- To introduce the concepts of analysis and design
- To model the system with various software diagrams
- To develop a system using software engineering concepts
- To prepare document/report of the system

Prerequisites:

Knowledge of Programming

Contents:

1. **Introduction to System Analysis & Design**
   Concepts of System Analysis & Design, Components of a System, Users of a System, System Development Life Cycle, Role of System Analyst, Various approaches to develop a system, Using a "systems" approach to build whole system, Applying concepts of Distributed Systems Development

2. **Structured Systems Development Approach**
   Requirements Gathering, surveying existing systems, Performing Feasibility Study before building a system, Formulating SRS, Designing Data Flow Diagrams, Entity-Relationship Diagrams, Designing Data Dictionary, Designing effective Input and Reports, Implementing EOD, EOM and EOY Procedures, Documenting the system

Guidelines for Project:

1. Projects can be created by a single student or in a group of 2-3 students.
2. The definition of project is to be submitted within 15 days of the starting of the semester.
3. The project should be free from plagiarism.
4. A student/group is required to report to their project guides on regular basis.
5. A student/group is required to present the project during internal exams.
6. It is advisable that the project must include different forms and reports.
7. It is necessary to prepare system diagrams, data dictionary as per the convention.
8. Project documentation should be prepared for the semester end examination.
9. In the semester end examination, the project should be presented to the examiners along with the documentation (report).

References:

1. Stevens, Comer D., Douglas E., “Internetworking with TCP/IP Vol. III”, Prentice Hall
2. Galbraith J., “Network Programming in Python: The Basic: A Detailed Guide to Python 3 Network Programming and Management”, BPB Publication
3. Neidinger M., “Python Network Programming Techniques”, Packt Publishing Ltd.
4. Harold E.R., “Java Network Programming”, O’Reilly Media

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Analyze and Design a system based on user requirements
- To develop a software system using software engineering concepts
- To prepare a document for the system

Course Name: Mobile Application Development

Course Code: MCSGE321

Objectives:

The main objective of this course is to enable students to

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

Prerequisites:

Knowledge of the Core Java Programming, Database concepts

Contents:

1. **Android platform**

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

   <u>Understanding the anatomy of an Android application</u>

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

   <u>Managing application resources</u>

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

   <u>Android user interface fundamentals</u>

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

2. **Android Programming**

   <u>Drawing and animations</u>

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

Working with files and directories

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

Databases and content providers

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

Working in background

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

Hardware sensors

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

Multimedia and wireless networking and telephony

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

Monetizing, promoting, and distributing applications

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

References:

1. Reto Meier, "Professional ANDROID Application Development", Wrox Publication
2. Lauren Darcey and Shane Conder, “Android Wireless Application Development”, Pearson Education,
3. Mark L Murphy, “Beginning Android”, Wiley India Pvt Ltd
4. Josh Skeen, David Greenhalgh , “Android Programming: The Big Nerd Ranch Guide”, Big Nerd Ranch Guides
5. John Horton, “Android Programming for beginners", Packt Publication

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

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

Course Name: Web Framework (Elective-1)

Course Code: MCSDE321 (1)

Objectives:

The aim of this course is

- To introduce Django Framework
- To acquaint students to setup & Configure Django Framework
- To enable students to develop Web Applications using Django

Prerequisites:

Knowledge of HTML, PHP Programming

Contents:

1. **Django Framework & Templates**

   Django Framework Design Principles, Install Django, Start a Django Project, Set Up a Database for a Django Project, Set Up Content: Understand Urls, Templates, and Apps, Set Up the Django admin Site

   <u>Django URLs and Views</u>

   Url Regular Expressions, Url Parameters, Extra Options, and Query Strings, Url Consolidation and Modularization, Url Naming and Namespaces, View Method Requests, View Method Responses, View Method Middleware, Middleware Flash Messages in View Methods, Class-Based Views

   <u>Django Templates</u>

   Django Template Syntax, Django Template Configuration, Create Reusable Templates, Built-In Context Processors, Custom Context Processors, Built-In Django Filters, Built-In Django Tags, Custom Filters

   <u>Jinja Templates</u>

   Jinja Advantages and Disadvantages, Transition to Jinja Templates from Django Templates, Jinja Template Configuration in Django, Create Reusable Jinja Templates, Jinja Globals: Access Data on All Jinja Templates, Like Django Context Processors, Jinja Built-In Statements/Tags and Functions (Like Django Template Tags), Jinja Built-In Filters and Tests (Like Django Filters), Custom Filters and Tests in Jinja, Jinja Extensions, Jinja Policies

2. **Django Application Management, Forms & Models**

   Django settings.py for the Real World, Set Up Static Web Page Resources - Images, CSS, JavaScript, Django Logging, Django Email Service, Debug Django Applications, Django Management Commands

   <u>Django Forms</u>

   Django Form Structure and Workflow, Django Form Processing: Initialization, Field Access, Validation and Error Handling, Django Form Field Types: Widgets, Options, and Validations, Set Up the Layout for Django Forms in Templates, Django Custom Form Fields and Widgets, Django Advanced Form Processing: Partial Forms, AJAX, and Files, Django Formsets

   <u>Django Models</u>

   Django Models and the Migrations Workflow, Django Model Data Types, Django Model Default and Custom Behaviors, Relationships in Django Models, Django Model

Transactions, Django Model Migrations, Django Model Database Tasks, Django Model Initial Data Setup, Django Model Signals, Django Models Outside of models.py, Django Models and Multiple Databases

3. Django Model Queries, Forms & Class Views

CRUD Single Records in Django Models, CRUD Multiple Records in Django Models, CRUD Relationship Records Across Django Models, Model Queries by SQL Keyword, Model Queries with Raw (Open-Ended) SQL, Model Managers

Django Model Forms & Class Views

Django Model Form Structure and Workflow, Create Django Model Forms, Django Model Form Options and Field Mapping, Django Model Forms with Relationships, Django Model Form Processing, Django Model Formsets, Class-Based Views with Models

REST Services

REST Services in Django, Django REST Framework Concepts and Introduction, Django REST Framework Security

4. Django User Management & Administration

Django User Management

Introduction to the Django User System, Permission Types, Permission Checks and Enforcement, User Authentication and Auto-Management, Custom User Model Fields, Custom Authentication Back Ends, User Management with Django allauth, Social Authentication with Django allauth

Django admin Management

Set Up Django Models in the Django admin, Django admin Read Record Options, Django admin Create, Update, Delete Record Options, Django admin Custom Page Layout, Data, and Behaviors, Django admin CRUD Permissions, Multiple Django admin Sites

References:

1. Rubio D., “Beginning Django Web Application Development and Deployment with Python”, Apress
2. Aratyan T. , “Building Django 2.0 Web Applications”, Packt
3. George N., “Mastering Django: Core”, Packt
4. Forcier J., Bissex P., Chun W., “Python Web Development with Django”, Addison-Wesley Professional

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Get started with the Django framework
- Use Django views, class-based views, URLs, middleware, forms, templates, and Jinja templates
- Take advantage of Django models, including model relationships, migrations, queries, and forms
- Leverage the Django admin site to get access to the database used by a Django project
- Deploy Django REST services to serve as the data backbone for mobile, IoT, and SaaS systems

----

Course Name: Artificial Intelligence (Elective-1)

Course Code: MCSDE321 (2)

Objectives:

The main objective of this course is to enable students to

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

Prerequisite:

Basic knowledge of Mathematical Logic

Contents:

1. **AI Fundamentals, Search Techniques**
   Defining Artificial Intelligence, AI: Early history, Overview of AI Application Areas, Defining AI techniques, Turing Test, Intelligent Agents
   <u>State Space Search & Heuristic Search Techniques:</u>
   Defining problems as State Space Search, Production Systems and Characteristics, Breadth First and Depth First Search, Hill Climbing Techniques, Problems and Solutions, Best First Search, Constraint Satisfaction Problems

2. **Knowledge Representation, Using Predicate Logic**
   <u>Knowledge Representation Techniques:</u>
   Semantic Nets, Partitioned Semantic Nets, Conceptual Dependency, Rules for Knowledge Representation, Advantages, Limitations
   <u>Using Predicate Logic:</u>
   Representing simple facts in logic, Prepositional Logic, FOPL, Computable functions and predicates, Procedural vs. Declarative knowledge, Forward Vs. Backward Chaining

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

4. **PROLOG**
   PROLOG concepts: Basic structures, Recursion, Fail and Cut predicates, Compound objects, List, Programming in Prolog

References:

1. Elaine Rich, Kevin Knight, Shivashankar B. Nair, “Artificial Intelligence”, 3rd Edition, McGraw Hill.

2. Stuart Russell and Peter Norvig “Artificial Intelligence-A modern Approach”, 2nd Edition, Pearson Education.
3. Townsend C., “Introduction to Turbo PROLOG”, BPB Publications

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Represent knowledge effectively
- Apply problem solving techniques and intelligent search techniques

----

Course Name: Foundations of Cryptography

Course Code: MCSDE321 (3)

Objectives:

The main objective of this course is to enable students to

- Understand the state-of-the-art in the area of Applied Cryptography
- Study the underlying principles for confidentiality, integrity and authentication
- Develop a base for advanced study of subjects pertaining to Network Security and Web Security
- Implement various open source tools related to network security

Prerequisite:

Fundamentals of Computers

Contents:

1. **Foundations, Encryption Algorithms**

   Basic Principles: Need of Information Security, Security Risks, Security Services, Fundamentals of Cryptosystems, Cryptosystem Security Assumptions, Breaking Cryptosystems, Historical Cryptosystems Monoalphabetic Ciphers, Historical Advances, Theoretical Vs. Practical Security: Theoretical Security, Practical Security

   Algorithms:

   Symmetric Encryption: Classification of Algorithms, Stream Ciphers, Block Ciphers, The Data Encryption Standard(DES), Variants of DES, The Advanced Encryption Standard(AES), Modes of Operation, Use of Symmetric Encryption

   Public-key Encryption:

   Public-key Cryptography, RSA, ElGamal & Elliptic Curve Cryptography, Comparison, Use of Public-key Cryptography

2. **Data Integrity, Digital Signatures, Entity Authentication & Cryptographic Protocols**

   Data Integrity:

   Different levels of Data Integrity, Hash Functions, Message Authentication Codes

   Digital Signature Schemes:

   Digital Signatures, Non-repudiation using Symmetric Techniques, Digital Signatures schemes based on RSA, Digital Signature schemes in practice

   Entity Authentication:

   Random Number Generation, Providing Freshness, Fundamentals of Entity Authentication, Passwords, Dynamic Password Schemes, Zero-knowledge mechanism

   Cryptographic Protocols:

   Protocol basics, Protocol design, Analyzing a simple protocol, Authentication & Key Establishment protocols

3. **Key Management**

   Key Management Principles:

   Fundamentals, Key Lengths and Lifetime, Key Generation, Key Establishment, Key Storage, Key Usage, Governing Key Management

   Public-key Management:

Certificate of Public-keys, The Certificate Lifecycle, Public-key management models, Alternative approaches

4. **Case Study of Applications of Cryptography & Open Source Tools**
Applications of Cryptography for (a) Internet Security (b) Wireless LAN (c) Secure Payment Card Transactions (d) Video Broadcasting (e) Identity Cards (f) Anonymity (g) Digital Currency (Bitcoin), Tools & Standards: OpenSSL, GPG, Kerberos, PKCS, etc.

References:

1. Martin K., "Everyday Cryptography – Fundamental Principles & Applications", 2nd Ed., Oxford University Press
2. Schneier B., "Applied Cryptography", John Wiley & Sons
3. Menzes A., Oorschot P., Vanstone S., "Handbook of Applied Cryptography", CRC Press
4. Stallings W., "Cryptography and Network Security", Pearson India
5. Forouzan B., "Cryptography and Network Security", Tata McGraw Hill
6. Howlett T., "Open Source Security Tools", Prentice Hall

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Apply the principles of cryptography in various areas of Network Security and Web Security
- Have thorough understanding of various standards and algorithms of Cryptography
- Effectively use various tools and utilities related to cryptography

----

Course Name: Web Framework (Elective-1) – Practical

Course Code: MCSDE322 (1)

Objectives:

The main objective of this course is to enable students to

- Perform hands-on on Django Framework
- Develop Web Applications using Django

Prerequisite:

Fundamentals of Computers

Contents:

1. **Django Framework & Templates**
   Setting up Django Environment, Working with URL Namespaces, URL Consolidation & Modularization, Implementing Method Requests & View Methods, Using Django Templates and Jinja Templates

2. **Django Application Management, Forms & Models**
   Configuring Django for Email Services and Logging, Django Management using commands, Implementing Forms using Django, Using Django Models, Performing Database operations in Django, Using Django Model Forms & Implementing views, Implementing REST services in Django, Performing various administrative tasks & user management in Django

References:

1. Rubio D., "Beginning Django Web Application Development and Deployment with Python", Apress
2. Aratyan T. , "Building Django 2.0 Web Applications", Packt
3. George N., "Mastering Django: Core", Packt
4. Forcier J., Bissex P., Chun W., "Python Web Development with Django", Addison-Wesley Professional

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Customize Django Framework
- Develop Web Applications using Django Framework
- Perform Administrative tasks using Django Framework

Course Name: Artificial Intelligence (Elective-1) – Practical

Course Code: MCSDE322 (2)

Objectives:

The main objective of this course is to enable students to

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

Prerequisite:

Basic knowledge of Mathematical Logic

Contents:

1. **Implementing Clauses & Predicates & Basics of AI Programming**
   Programming environment for Artificial Intelligence, Using Clauses & Predicates, Loading clauses, variables, Unification, Evaluating Goals, Backtracking, Using Operators, Inputting terms & characters, Outputting terms & characters, Input and Output using Files, Implementing loops, Recursion, Backtracking with failure, Preventing Backtracking, Changing the Database

2. **Knowledge Representation, Using Predicate Logic**
   Processing Lists, Processing Strings, Using Grammar Rules to analyze English Sentences, Implementing an Artificial Language, Developing an Expert System Shell

References:

1. Bramer M., “Logic Programming with Prolog”, Springer
2. Townsend C., “Introduction to Turbo PROLOG”, BPB Publications
3. Bonaccorso et. al., “Python: Advanced Guide to Artificial Intelligence: Expert machine learning systems and intelligent agents using Python”, Packt Publications
4. Shaffer S., “Code your own rule-based expert system”, Shaffer Media Enterprises
5. Elaine Rich, Kevin Knight, Shivashankar B. Nair, “Artificial Intelligence”, 3rd Edition, McGraw Hill
6. Stuart Russell and Peter Norvig “Artificial Intelligence-A modern Approach”, 2nd Edition, Pearson Education

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Implementing fundamentals of AI using programming language
- Build effective small scale expert systems

Course Name: Foundations of Cryptography – Practical

Course Code: MCSDE322 (3)

Objectives:

The main objective of this course is to enable students to

- Understand the state-of-the-art in the area of Applied Cryptography
- Study the underlying principles for confidentiality, integrity and authentication
- To develop a base for advanced study of subjects pertaining to Network Security and Web Security
- To implement various open source tools related to network security

Prerequisite:

Fundamentals of Computers

Contents:

1. **Foundations, Encryption Algorithms**
   Generate private and public keys, Generate Random numbers, Implement or Use basic cryptographic algorithms like DES, AES, RSA, Elgamal etc. using state of art tools over strings and on files. Applying different cipher methods. Use tools like Wireshark and nmap to perform protocol analysis

2. **Data Integrity, Digital Signatures, Entity Authentication & Cryptographic Protocols**
   Implement or Use message digest algorithms like MD5, SHA to generate message digests on strings and on files. Generate signatures, certificates, view and install certificates

References:

1. Martin K., “Everyday Cryptography – Fundamental Principles & Applications”, 2nd Ed., Oxford University Press
2. Schneier B., “Applied Cryptography”, John Wiley & Sons
3. Menzes A., Oorschot P., Vanstone S., “Handbook of Applied Cryptography”, CRC Press
4. Stallings W., “Cryptography and Network Security”, Pearson India
5. Forouzan B., “Cryptography and Network Security”, Tata McGraw Hill
6. Howlett T., “Open Source Security Tools”, Prentice Hall

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Apply various algorithms and techniques using state of art tools
- Implement Data Integrity & Digital Signatures using tools
- Effectively use various tools and utilities related to cryptography

----

5 Years Integrated MSc (Computer Science)

[Course Initiated in 2021-2023]

Semester - VII

Scheme & Course Structure

Applicable from July 2023

Course Structure with Credits

	Course Type	Subject Names & Credits	Total Credits
Semester-VII	Core Courses	• MCSCC411-System Software (4) • MCSCC412-Computer Oriented Numerical Methods (4) • MCSCC413- Computer Oriented Numerical Methods - Practical (2)	**10**
	Ability Enhancement Courses	• MCSAE411 – Data Visualization (2)	**2**
	Skill Enhancement Courses	• MCSSE411 - Project-5 (4)	**4**
	Discipline Specific Elective Courses	• MCSDE411 - Elective-2 (4) • MCSDE412 - Elective-2 – Practical (2)	**6**
	Generic Elective Courses	• MCSGE411 - Innovation and Startup (2)	**2**
	Semester-VII (Total Credits)		**24**

TEACHING & EXAMINATION SCHEME FOR 5 YEAR INTEGRATED M. Sc. (COMPUTER SCIENCE) COURSE

Semester-VII	Course		Hours per week								Credits
	No.	Name	Theory	Practical	Internal	Total	Lectures	Others (Tutorials)	Practical	Total	
Semester-VII	MCSCC411	System Software	70	***	30	100	3	1	***	4	4
	MCSCC412	Computer Oriented Numerical Methods	70	***	30	100	3	1	***	4	4
	MCSCC413	Computer Oriented Numerical Methods - Practical	***	35	15	50	***	***	4	4	2
	MCSAE411	Data Visualization	***	35	15	50	***	***	4	4	2
	MCSSE411	Project – 5	***	70	30	100	***	***	8	8	4
	MCSDE411	Elective-2	70	***	30	100	3	1	***	4	4
	MCSDE412	Elective-2 – Practical	***	35	15	50	***	***	4	4	2
	MCSGE411	Innovation and Startup	35	***	15	50	2	***	***	2	2
		Total	245	175	180	600	11	3	20	34	24

Theory/Tutorial Credits → 1 Hour = 1 Credit

Practical Credits → 2 Hours = 1 Credit

Track wise subjects for Elective-2			
Track	**Web Technologies**	**Artificial Intelligence & Machine Learning**	**Security**
Subject Code	MCSDE411(1)	MCSDE411(2)	MCSDE411(3)
Subject Name	Full Stack Web Development	Data Analytics	Network Security
Subject Code	MCSDE412(1)	MCSDE412(2)	MCSDE412(3)
Subject Name	Full Stack Web Development - Practical	Data Analytics – Practical	Network Security - Practical

Course Name: System Software

Course Code: MCSCC411

Objectives:

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

- Understand the relationship between system software and machine architecture
- Know the design and implementation of assemblers, macro processor, linker and compiler
- Have an understanding of loader, system software tools
- Understand and know the working of device drivers

Prerequisites:

Basic knowledge of computer architecture and C programming

Contents:

1. **Introduction to System Software and software tools Language Processors**
   Introduction, Language Processing Activities, Fundamentals of Language Processing & Language Specification, Language Processor Development Tools. Data Structures for Language Processing: Search Data structures, Allocation Data Structures. Software Tools: Software Tools for Program Development, Editors, Debug Monitors, Programming Environments, and User Interfaces

2. **Assemblers & Macro Processors**
   Assemblers: Elements of Assembly Language Programming, A Simple Assembly Scheme, Pass Structure of Assemblers, Design of a Two Pass Assembler, A single pass Assembler for IBM PC.
   Macros and Macro Processors: Macro Definition and Call, Macro Expansion, Nested Macro Calls, Advanced Macro Facilities, Design of a Macro Preprocessor

3. **Compilers and Interpreters**
   Scanning and Parsing: Introduction to NFA and DFA, Approaches of parsing, Different types of parsing techniques namely, Recursive decent parser, LL(1) parser, Operator precedence parser. (First and follow technique for generating a parse table is to be taught), Phases of the Compiler, Aspects of compilation, Memory allocation. Compilation of expressions and control structures. Code optimization. Interpreters: Use and overview of interpreters, pure and impure interpreters

4. **Linkers, Loaders and Device Drivers**
   Introduction to linkers, Relocation and Linking Concepts. Design of a Linker, Self-Relocating Programs. A Linker for MS-DOS, Linking for Overlays. Introduction to Loaders, Design and anatomy of UNIX device driver, Types of device driver, General design of UNIX character device driver, General design of UNIX block device driver, UNIX device driver installation

References:

1. Dhamdhere D.M., “Systems Programming and Operating Systems”, Tata McGraw Hill
2. Beck L., “System Software – An introduction to systems programming”, Pearson Education
3. Pajari G., “Writing UNIX device drivers”, Pearson Education
4. Chattopadhyay S., “System Software”, Prentice Hall India
5. Aho A., Ulman J., Lam M., Sethi R., “Compilers: Principles, Techniques and Tools”, Pearson Education
6. Louden K., “Compiler Construction: Principles and Practice”, Cengage India

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Good understanding of basic design and implementation of various system software
- Ability to design and implement better and more efficient programs/applications as a result of the knowledge gained

----

Course Name: Computer Oriented Numerical Methods

Course Code: MCSCC412

Objectives:

The main objective of this course is to enable students to

- Understand numerical methods for problems of numerical analysis
- Gain experience in the implementation of numerical methods using computers
- Appreciate the concept of error and the need to analyze and predict it

Prerequisites:

Foundations in Mathematics, Programming in C/C++

Contents:

1. **Introduction to Numerical Methods**

   Characteristic of Numerical Methods, Different Sources of Errors, Iterative Methods, Types of Errors, Quantification of Errors, Stopping Criterion in iterative methods

2. **Iterative Methods for finding roots of an equation f(x) = 0**

   Bisection ,False Position, Secant, Newton Raphson and Successive Approximation Method, Descarte's rule of sign, Budan's Theorem, Birge Vieta Method

3 **Polynomial Interpolation, Least Square Approximation**

Lagrange, forward difference, backward difference, divided difference interpolation, Error Estimates, Cubic spline interpolation, Least square Curve fitting: Linear Regression and Nonlinear Regression, Approximation of Functions by Taylor series, Chebyshev Approximation, Chebyshev Economization

4 **Numerical Differentiation, Integration and Solution of Ordinary Differential Equations**

Differentiation, Numerical Integration – Newton Cotes formulas, Trapezoidal, Simpson's 1/3 & 3/8 rules, Gauss Quadrature formulas, Taylor series, Euler, Runge-Kutta 2nd order and 4th order, Predictor Corrector methods: Milne Simpson & Adam's Moulton

References:

1. Vachharajani V., "Numerical Analysis – A programming approach", BPB Publications
2. Gerald C.F., Wheatley P.O., "Applied Numerical Analysis", Pearson Education
3. Chapra S., Canale R., "Numerical Methods for Engineers", Tata McGraw Hill Pub.
4. Somasundaram R., Chandrasekaran R., "Numerical Methods with C++ Programming", Prentice Hall Pvt. Ltd.
5. Datta N., "Computer Oriented Numerical Methods", Vikas Publications
6. Burden R., Faires D., "Numerical Analysis", Cengage Pubilcation
7. Mathews J., Fink K., "Numerical Methods Using MATLAB", Pearson Education

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Gain ability to solve linear and non-linear algebraic equations, perform operations of calculus, fit curves, and solve differential equations using a computer
- Appreciate problems due to rounding errors and convergence

Course Name: Computer Oriented Numerical Methods – Practical

Course Code: MCSCC413

Objectives:

The objective of the course is to enable students to

- Gain experience in the implementation of numerical methods using computers
- Appreciate the concept of error and the need to analyze and predict it

Prerequisites:

Foundations in Mathematics, Programming in C/C++

Contents:

1. **Finding Roots, Polynomial Interpolation**

   Implementation of Bisection Method, False Position Method, Secant Method, Newton Raphson Method and Successive Approximation Method, Validation of Descarte's rule of sign, Implementation of Birge Vieta Method, Implementation of Lagrange forward difference method, backward difference method, divided difference interpolation, Performing Error Estimates, Cubic spline interpolation, Implementing Least square Curve fitting: Linear Regression and Nonlinear Regression, Approximation of Functions by Taylor series, Implementation of Chebyshev Approximation

2. **Numerical Differentiation, Integration and Solution of Ordinary Differential Equations**

   Implementation of various Differentiation methods & Numerical Integration methods like – Newton Cotes formulas, Trapezoidal rule, Simpson's 1/3 & 3/8 rules, Gauss Quadrature formulas, Programming Taylor series, Euler, Runge-Kutta 2nd order and 4th order techniques, Implementing Predictor Corrector methods

References:

1. Vachharajani V., "Numerical Analysis – A programming approach", BPB Publications
2. Gerald C.F., Wheatley P.O., "Applied Numerical Analysis", Pearson Education
3. Chapra S., Canale R., "Numerical Methods for Engineers", Tata McGraw Hill Pub.
4. Somasundaram R., Chandrasekaran R., "Numerical Methods with C++ Programming", Prentice Hall Pvt. Ltd.
5. Datta N., "Computer Oriented Numerical Methods", Vikas Publications
6. Burden R., Faires D., "Numerical Analysis", Cengage Pubilcation
7. Mathews J., Fink K., "Numerical Methods Using MATLAB", Pearson Education

Accomplishments of the student after completing the course:

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

- Solve linear and non-linear algebraic equations, perform operations of calculus, fit curves, and solve differential equations using a computer
- Appreciate problems due to rounding errors and convergence

Course Name: Data Visualization

Course Code: MCSAE411

Objectives:

The main objective of this course is to enable students to

- Gain knowledge of web data scraping
- Apply pre-processing on data
- Deliver data to web applications
- Generate efficient charts and plots

Prerequisites:

Fundamentals of Javascript, Fundamentals of Python

Contents:

1. **Data Processing with Python**
   Development Setup, Reading & Writing data in Python, Getting data off the web with Python, Scraping with Scrapy, Using Numpy, Using pandas, Cleaning data with pandas, Visualizing data with Matplotlib, Exploring data with pandas

2. **Delivering the Data**
   Delivering the data, RESTful data with Flask, Bringing charts to the web with Matplotlib & Plotly, A noble visualization, Building a Visualization, Using D3, Visualizing individual Prizes, Mapping with D3, Visualizing individual Winners, The Menu Bar

References:

1. Dale Kyran, “Data Visualization with Python and Javascript”, O’Reilly Media
2. Camm, Cochran, et. al., “Data Visualization: Exploring and Explaining the Data”, Cengage
3. Knaflic C.N., “Storytelling with Data”, Wiley
4. Teller S., “Data Visualization with d3.js”, Packt Publishing
5. Dixit R., “Data Analysis with Python”, BPB Publications
6. Hajba G.L., “Website Scrapping with Python”, Apress

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Scrap data from web and apply preprocessing
- Create visualizations to be delivered on Web and for analytics

----

Course Name: Project - 5

Course Code: MCSSE411

Objectives:

The aim of this course is

- To introduce structured development of software systems
- To acquaint students to various techniques of requirements determination
- To introduce the concepts of analysis and design
- To model the system with various software diagrams
- To develop a system using software engineering concepts
- To prepare document/report of the system

Prerequisites:

Knowledge of Programming

Contents:

1. **Introduction to System Analysis & Design**
   Concepts of System Analysis & Design, Components of a System, Users of a System, System Development Life Cycle, Role of System Analyst, Various approaches to develop a system, Using a "systems" approach to build whole system,

2. **Structured Systems Development Approach**
   Requirements Gathering, surveying existing systems, Performing Feasibility Study before building a system, Formulating SRS, Designing Data Flow Diagrams, Entity-Relationship Diagrams, Designing Data Dictionary, Designing effective Input and Reports, Implementing EOD, EOM and EOY Procedures, Performing Testing for the System, Evaluating Security Risks for the System, Documenting the System

Guidelines for Project:

1. Projects can be created by a single student or in a group of 2-3 students.
2. The definition of project is to be submitted within 15 days of the starting of the semester.
3. The project should be free from plagiarism.
4. A student/group is required to report to their project guides on regular basis.
5. A student/group is required to present the project during internal exams.
6. It is advisable that the project must include different forms and reports.
7. It is necessary to prepare system diagrams, data dictionary as per the convention.
8. Project documentation should be prepared for the semester end examination.
9. In the semester end examination, the project should be presented to the examiners along with the documentation(report).

References:

1. Kendall & Kendall. "System Analysis and Design", Prentice Hall India (PHI)
2. James Sen, "Analysis and Design of Information System", PHI

3. Arthur Langer, “Analysis and Design of Information Systems“, Springer
4. Jeff Johnson, “Designing with the Mind in Mind“, Morgan Kaufmann
5. Dennis, Wixom, Roth “System Analysis and Design”, Wiley India
6. Joseph Valacich, Joey F. George, “Modern Systems Analysis and Design”, Pearson
7. Priti Srinivas Sajja., “Essence of System Analysis and Design”, Springer

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Analyse and Design a system based on user requirements
- Develop a software system using software engineering concepts
- Prepare a document for the system

Course Name: Full Stack Web Development

Course Code: MCSDE411 (1)

Objectives:

The objective of the course is to enable students to

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

Prerequisites:

Knowledge of HTML, DHTML and Object-Oriented programming

Contents:

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

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

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

4. **Dependency Injection, Services, Routing and Navigation & NodeJS**
   Dependency injection, Using dependency injection, Angular JS service –Understanding services, Using built-in service, Creating custom service, Injecting dependency in service, Overview of Routing, Routing using ngRoute and UI- Router, ngView Directive, Configuring $routeProvider, $stateProvider, Animating Angular App, Overview of TypeScript, Introduction to Node.js, Features of Node.js, Setup Development Environment- Installing Node.js, Working with REPL, Node.js Console, Node.js Module, Node Package Manager, Node.js Basics, File System, HTTP and HTTPs, Creating Web Server- Handling http request, Node.js Callbacks, Node.js Events

References:

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

Accomplishments of the student after completing the course:

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

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

Course Name: Data Analytics

Course Code: MCSDE411 (2)

Objectives:

The objective of the course is to enable students to

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

Prerequisites:

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

Contents:

1. **Introduction**

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

2. **Descriptive statistics**

   Graphical & Tabular methods

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

   Numerical methods

   Measures of location: Mean median, mode, percentiles, quartiles,

   Measures of variability: range, interquartile range, variance, standard deviation, coefficient of variation,

   Measures of association between two variables: covariance interpretation of the covariance correlation coefficient interpretation of the correlation coefficient, Measures of distribution: shape, relative location, and detecting outliers z-scores empirical rule detecting outliers box plot, Handling descriptive statistics through python programs and/or libraries

3. **Regression models, Probability & Probability Distribution**

Regression Models

Simple and Multiple linear regression model, least squares method, coefficient of determination model assumptions testing for significance, estimate of σ2, t-test, and confidence interval. Implementing regression models using Sci-kit library

Probability and Probability distribution

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

4. **Hypothesis Testing, Time Series & Forecasting**

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

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

References:

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

Accomplishments of the student after completing the course:

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

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

----

Course Name: Network Security

Course Code: MCSDE411 (3)

Objectives:

The objective of the course is to enable students to

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

Prerequisites:

Basics of Computer Networking

Contents:

1. **Network Security Fundamentals, Issues and Challenges**

   Computer Network Security Fundamentals

   Introduction, Securing the Computer Network, Forms of Protection, Security Standards

   Security Threats and Threat Motives to Computer Networks

   Sources of Security Threats, Security Threat Motives, Security Threat Management, Security Threat Correlation, Security Threat Awareness

   Computer Network Vulnerabilities

   Definition, Sources of Vulnerabilities, Vulnerability Assessment

   Cybercrimes and Hackers

   Introduction, Cybercrimes, Hackers, Dealing with Cybercrimes

   Security Assessment, Analysis, and Assurance

   System Security Policy, Building a Security Policy, Security Requirements Specification, Threat Identification, Threat Analysis, Vulnerability Identification and Assessment, Security Certification, Security Monitoring and Auditing, Products and Services

2. **Cryptography & Access Control, Authorization and Authentication**

   Cryptography

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

   Access Control and Authorization

   Definitions, Access Rights, Access Control Systems, Authorization, Types of Authorization System, Authorization Principles, Authorization Granularity, Web Access and Authorization

   Authentication

   Definition, Multiple Factors and Effectiveness of Authentication, Types of Authentication, Authentication Methods, Developing an Authentication Policy

3. **Firewalls, IDS and IPS**

   Firewalls

Definition, Types of Firewalls, Configuration and Implementation of a Firewall, The Demilitarized Zone (DMZ), Improving Security Through the Firewall, Firewall Forensics, Firewall Services and Limitations

Intrusion Detection & Prevention

Definition, Intrusion Detection, Intrusion Detection Systems (IDSs), Types of Intrusion Detection Systems, The Changing Nature of IDS Tools, Other Types of Intrusion Detection Systems, Response to System Intrusion, Challenges to Intrusion Detection Systems, Implementing an Intrusion Detection System, Intrusion Prevention Systems (IPSs), Intrusion Detection Tools

4. **Dealing with Computer Network Security Challenges**

Computer and Network Forensics

Introduction, Computer Forensics, Network Forensics, Forensic Tools

Virus and Content Filtering

Definitions, Scanning, Filtering, and Blocking, Virus Filtering, Content Filtering, Spam

Standardization and Security Criteria

Introduction, Product Standardization, Security Evaluations, Major Security Evaluation Criteria

Computer Network Security Protocols

Introduction, Application-Level Security, Security in the Transport Layer, Security in the Network Layer, Security in the Physical Layer

Security Beyond Wired Networks

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

Case Study of Open Source Tools

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

References:

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

Accomplishments of the student after completing the Course:

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

- Analyze and develop solutions to solve network security related problems.
- Have thorough understanding of various standards and protocols related to computer network security

- Effectively use various tools and utilities for computer networking.

Course Name: Full Stack Web Development - Practical

Course Code: MCSDE412 (1)

Objectives:

The objective of the course is to enable students to

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

Prerequisites:

Knowledge of HTML, DHTML and Object-Oriented programming

Contents:

1. **Java Script, JSON & AngularJS**

   Java Script

   Basic Programming in Java Script, Use of Primitives, Operations and Expressions, Screen Output and Keyboard Input, Control Statement, Object Creation and modification, Arrays, Functions, Constructors, Pattern Matching using regular expressions, Creating XML and JSON documents, Implementation of Callback functions, Promise Objects, async…await

   AngularJS

   Implementing simple programs using AngularJS, Using MVC architecture, Implementing expression and data binding, Using directives, Using Angular Modules, Creating Controller, Scope, View and Modules, Using AngularJS forms, Applying various filters and formatting, Using AngularJS services, Creating custom services

2. **WebSocket & NodeJS**

   Ajax implemention using $http and Web APIs, Implementation of AJAX programs, Implementation of WebSockets, Using dependency injection, Setting up NodeJS environment, Using Node.js console, Using Node.js module, Using Node Package Manager, Implementing programs that uses file system, Using Node.js callbacks, Using Node.js events, Creating webserver and handling requests

References:

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

Accomplishments of the student after completing the course:
At the end of the course, students will be able to

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

Course Name: Data Analytics – Practical

Course Code: MCSDE412 (2)

Objectives:

The objective of the course is to enable students to

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

Prerequisites:

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

Contents:

1. **Data Analytics Tools & Technologies**
   Features of R, Python for Data Analytics, Using Pandas & Matplotlib for Data Analytics, Data extraction and cleaning, Data transformation and normalization, Data visualization and exploration

2. **Descriptive & Diagnostic Analysis**
   Measures of central tendency (mean, median, mode), Measures of dispersion (variance, standard deviation, range), Boxplots and histograms, Correlation analysis, Regression analysis, Outlier detection

References:

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

Accomplishments of the student after completing the course:

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

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

----

Course Name: Network Security - Practical

Course Code: MCSDE412 (3)

Objectives:

The objective of the course is to enable students to

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

Prerequisites:

Basics of Computer Networking

Contents:

1. **Cryptography Tools:**

   NMAP & Wireshark

   Scanning the network using Nmap for open ports, Analyzing the protocols and network using Wireshark, Using various filters of Wireshark

   OpenSSL

   Setting up environment for OpenSSL, Generating random passwords with OpenSSL, Generating Public and Private Key using OpenSSL, Encrypting files and strings using DES, 3DES, AES, RSA algorithm, Using different cipher modes in OpenSSL, Creating Message Digests using OpenSSL, Creating and Installing certificates using OpenSSL

2. **Firewalls & IDS/IPS**

   Firewall

   Enabling/Disabling a Firewall, Setting up rules for firewall to drop packets, Setting up rules for firewall to accept packets

   IDS/IPS

   Setting up environment for IDS/IPS, Applying signatures, Using sniffer mode, Using packet logger mode, Using NIDS mode, Implementing snort rules format, using snort command line options, Setting up alert rules

   OSSEC and OpenVAS

   Overview and demonstration of OSSEC and OpenVAS tools

References:

1. Kizza J.M., “Guide to Computer Network Security”, 4th Ed., Springer International Publishing
2. Kaufman C., Perlman R., Speciner M., “Network Security: Private Communication in Public World”, Prentice Hall India
3. Perez A., “Network Security”, John Wiley & Sons

4. Stallings W., “Network Security Essentials – Applications and Standards”, Pearson India
5. Forouzan B., “Cryptography and Network Security”, Tata McGraw Hill
6. Howlett T., “Open Source Security Tools”, Prentice Hall

Accomplishments of the student after completing the Course:
At the end of the course, students will be able to

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

----

Course Name: Innovation and Startup

Course Code: MCSGE411

Objectives:

The main objective of this course is to enable students to

- Gain knowledge regarding building a startup
- Prepare an organization for Innovation
- Apply innovation in startups

Prerequisites:

Fundamentals of Computers

Contents:

1. **Building a Startup**
   Starting with an Idea, Making a Business Plan, Securing Funds for Startup, Identifying the right people to be Involved, Following Legal Steps, Setting up a Location, Developing a Marketing Plan, Setting up a Customer Base, Preparing the organization for Innovation, Promoting and Communication, Personal creativity for Entrepreneurs, Corporate Creativity and Innovation, Innovation theory for Entrepreneurs, Business Essentials for Innovative Entrepreneurs, Innovation Readiness and Deployment

2. **Innovation and Startup**
   Innovation Process Model, Innovation and Consumer/Customer Insight, Building a strong Organizational Foundation for Innovation, Innovation Methods, Managing an Intellectual Property Portfolio, Importance of STEM and TRIZ in an Innovation Infrastructure, Global Innovation Problem Areas and Quality of R&D Ideas, Total Innovation Management for Excellence (TIME)

References:

1. Nager M., Nelsen C., Nouyrigat Y., “Startup Weekend: How to Take a Company From Concept to Creation in 54 Hours”, John Wiley & Sons
2. Fernandex et. al., “The framework for innovation: an entrepreneur’s guide to the body of Innovation Knowledge”, CRC Press
3. Blank S., Dorf B., “The Startup Owner’s Manual”, John Wiley & Sons
4. Kennard M., “Innovation and Entrepreneurship”, Routledge

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Build their own startup
- Sustain the startup with innovations

----

5 Years Integrated MSc (Computer Science)

[Course Initiated in 2021-2023]

Semester - VIII

Scheme & Course Structure

Applicable from July 2023

Course Structure with Credits

	Course Type	Subject Names & Credits	Total Credits
Semester-VIII	Core Courses	• MCSCC421-Cloud Computing (4) • MCSCC422-Application Development Project (8)	**4**
	Ability Enhancement Courses		**-**
	Skill Enhancement Courses		**8**
	Discipline Specific Elective Courses	• MCSDE421-Elective-3 (4) • MCSDE422-Elective-4 (4) • MCSDE423-Elective-3 – Practical (2) • MCSDE424-Elective-4 – Practical (2)	**12**
	Generic Elective Courses		**-**
	Semester-VIII (Total Credits)		**24**

TEACHING & EXAMINATION SCHEME FOR 5 YEAR INTEGRATED M. Sc. (COMPUTER SCIENCE) COURSE

Semester-VIII	Course		Hours per week								Credits
	No.	Name	Theory	Practical	Internal	Total	Lectures	Others (Tutorials)	Practical	Toal	
Semester-VIII	MCSCC421	Cloud Computing	70	***	30	100	3	1	***	4	4
	MCSCC422	Application Development Project	***	140	60	200	***	2	12	14	8
	MCSDE421	Elective-3	70	***	30	100	3	1	***	4	4
	MCSDE422	Elective-4	70	***	30	100	3	1	***	4	4
	MCSDE423	Elective-3 - Practical	***	35	15	50	***	***	4	4	2
	MCSDE424	Elective-4 – Practical	***	35	15	50	***	***	4	4	2
		Total	210	210	180	600	9	5	20	34	24

Theory/Tutorial Credits → 1 Hour = 1 Credit, Practical Credits → 2 Hours = 1 Credit

Track wise subjects for Elective-3 and Elective-4			
Track	**Web Technologies**	**Artificial Intelligence & Machine Learning**	**Security**
Subject Code	MCSDE421(1)	MCSDE421(2)	MCSDE421(3)
Subject Name	Web Server Management	Machine Learning	Blockchain Technology
Subject Code	MCSDE423(1)	MCSDE423(2)	MCSDE423(3)
Subject Name	Web Server Management - Practical	Machine Learning - Practical	Blockchain Technology - Practical
Subject Code	MCSDE422(1)	MCSDE422(2)	MCSDE422(3)
Subject Name	Web Data Management	Deep Learning	Internet of Things
Subject Code	MCSDE424(1)	MCSDE424(2)	MCSDE424(3)
Subject Name	Web Data Management - Practical	Deep Learning - Practical	Internet of Things - Practical

Course Name: Cloud Computing

Course Code: MCSCC421

Objectives:

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

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

Pre-requisites:

Basic Knowledge of Operating Systems, Knowledge of Object-Oriented Programming Language, Knowledge of Database Systems

Contents:

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

2. **Virtualization, Scaling, Capacity Planning & Cloud Technologies**
   Resource Virtualization, Approaches to virtualization, Hypervisors, Resource pooling, Resource Sharing and Provisioning, Scaling in the cloud, Capacity Planning and Load Balancing
   The File system and Storage services, Database technologies, Networking technologies over cloud & CDN, Cloud Management and Programming Model, Security reference model, Security aspects, Platform related security, Audit and Compliance, Portability and Interoperability issues

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

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

References:

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

Accomplishments of the student after completing the Course:
At the end of the course, students will be able to

- Decide and implement appropriate cloud computing infrastructure for any organization
- Design customized programming solutions using cloud computing model

----

Course Name: Application Development Project

Course Code: MCSSE421

Objectives:

The aim of this course is

- To introduce structured development of software systems
- To acquaint students to various techniques of requirements determination
- To introduce the concepts of analysis and design
- To model the system with various software diagrams
- To develop a complete system/module of a large system using software engineering concepts
- To prepare document/report of the system

Prerequisites:

Knowledge of Programming

Contents:

1. **Introduction to System Analysis & Design**
   Concepts of System Analysis & Design, Components of a System, Users of a System, System Development Life Cycle, Role of System Analyst, Various approaches to develop a system, Using a "systems" approach to build whole system

2. **Structured Systems Development Approach**
   Requirements Gathering, surveying existing systems, Performing Feasibility Study before building a system, Formulating SRS, Designing Data Flow Diagrams, Entity-Relationship Diagrams, Designing Data Dictionary, Designing effective Input and Reports, Implementing EOD, EOM and EOY Procedures, Documenting the system

Guidelines for Project:

1. Projects can be created by a single student or in a group of 2-3 students.
2. The definition of project is to be submitted within 15 days of the starting of the semester.
3. The project should be free from plagiarism.
4. A student/group is required to report to their project guides on regular basis.
5. A student/group is required to present the project during internal exams.
6. It is advisable that the project must include different forms and reports.
7. It is necessary to prepare system diagrams, data dictionary as per the convention.
8. Project documentation should be prepared for the semester end examination.
9. In the semester end examination, the project should be presented to the examiners along with the documentation (report).

References:

1. Pressman R., Maxim B., "Software Engineering – A practioner's approach", McGraw Hill
2. Kendall & Kendall. "System Analysis and Design ", Prentice Hall India (PHI)

3. James Sen, “ Analysis and Design of Information System”, PHI
4. Arthur Langer, “ Analysis and Design of Information Systems “, Springer
5. Jeff Johnson, “ Designing with the Mind in Mind “, Morgan Kaufmann
6. Dennis, Wixom, Roth “System Analysis and Design”, Wiley India
7. Joseph Valacich, Joey F. George, “Modern Systems Analysis and Design”, Pearson
8. Priti Srinivas Sajja., “Essence of System Analysis and Design”, Springer

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Analyse and Design a system based on user requirements
- Develop a software system using software engineering concepts
- Prepare a document for the system

Subject Name: Web Server Management

Subject Code: MCSDE421 (1)

Objectives:

The aim of this course is

- To enable students to know the concept of web server management and apply the concepts in practical

Prerequisite:

Basics of Linux

Contents:

1. **Installation, Adding Common Modules**
   Install Apache on Linux and Windows, Debian Packages, Subversion Sources, Apache Toolbox, Starting and stopping Apche, Restarting Apache, Apache uninstallation, Which Apache versions Version of Apache to Use, upgrades, config.nice, boot, Useful configure script, options configure Options, Finding Apache's files location Files, Installing a Generic installation modules, third-party modules, Installing Unix mod_dav, Installing Windows mod_dav, Installing installation mod_perl, Unix Unix mod_perl, Installing Unix mod_php, Installing Windows mod_php, Installing installation mod_ssl, finding Modules Using modules locating, modules.apache.org, Installing mod_security,

2. **Logging, Virtual Hosts, Aliases, Redirecting, and Rewriting**
   Log entries, logs error, POST Contents, IP addresses and clients, MAC addresses, Cookies, Local Pages, logs rotating, Hostnames, Maintaining logs virtual hosts, Proxy Requests, IP addresses servers, Referring Page, browser, software name, logs arbitrary request, logs arbitrary response, MySQL database logs, Syslog, User Directories, Setting Up Name-Based Virtual Hosts, default virtual hosts, address-based virtual hosts, virtual hosts address-based mixing with name-based, Mass Virtual Hosting with mod_vhost_alias, Mass Virtual Hosting Using Rewrite Rules, Logging for virtual hosts, Logfile, port-based virtual hosts, Displaying the Same Content on Several Addresses, Virtual Hosts in a Database, Directories URLs mapping, Creating a URLS, URLs aliasing, directories URLs mapping to same CGI directory, CGI directories creating, Redirecting URLs to another location, Permitting case-insensitive URLs, Showing PHP source, URLs (Uniform Resource Locators) replacing text, CGI arguments, requests unreferred denying access, query strings, SSL (Secure Socket Layers) redirecting, directories turning into hostnames, redirecting to single host, arguments, URLs (Uniform Resource Locators) elements, directories rewriting, query arguments, Using AliasMatch

3. **Security, SSL & Dynamic Content**
   Authentication system, Single-Use Passwords, Expiring Passwords, Limiting Upload Size, Images from Being Used Off-Site, requiring Strong Authentication, .htpasswd Files, Digest Authentication, Security in a Subdirectory, Lifting Restrictions, Selectively, files ownership, MySQL databases User Credentials, authentication usernames, authentication passwords, brute-force password attacks, Digest authentication, Credentials Embedded in URLs, WebDAV security, WebDAV enabling, proxies URL

access, files wrappers, module sets, Web Root, Restricting Range Requests, DoS attacks, mod_security, Chrooting Apache with mod_security, authentication migrating, Mixing read-only access, Using redirecting permanent, Installing SSL, Windows SSL installation, SSL Certificates , SSL (Secure Socket Layers) CAs, Serving a SSL, Authenticating with Client authentication SSL, virtual hosts SSL, SSL certificates wild card, Enabling a CGI Directory, CGI scripts enabling, CGI directories default documents, Using CGI programs launching, Using CGI scripts extensions, CGI testing setup, CGI form parameters, SSIs (Server-Side Includes) , Displaying date, last modified date, Including a Standard Header, Including the CGI program output, CGI scripts, CPAN, mod_perl Handler, PHP script handling, PHP installation, CGI parsing output, Parsing SSIs, Getting Perl scripts, Enabling Python Script Handling

4. Error Handling, Proxies, Performance & Directory Listings

A error handling host fields, CGI Scripts , Customized Error Messages, Error Documents in Multiple Languages , error handling redirecting, Internet Explorer Display Your Error Page , Notification on Error Conditions, Securing Your Proxy Server, Open Mail Relay, Forwarding Requests to Another Server, Blocking Proxied Requests, mod_perl proxying content, caching proxy servers, Filtering Proxied Content, authentication Proxied Server, proxies, mod_proxy_balancer, proxies virtual hosts, FTP, How Much Memory You Need, Benchmarking Apache with ab, KeepAlive Settings, Getting a Snapshot of Your Site’s Activity, DNS lookups, Symbolic Links, .htaccess Files, Content Negotiation, Optimizing Process Creation, Thread Creation, Caching, load balancing, Caching Directory Listings, Speeding Up mod_perl, Dynamic Content, Directory/Folder Listings, Header and Footer on Directory Listings, Applying a Style sheet, Hiding Things from the Listing, Searching for Certain Files in a Directory Listing, directories sorting, Client-Specified Sort Order, List directories formatting, client-specified formatting, Adding Descriptions to Files, Auto generated Document Titles, Changing the icons, Listing the Directories First, files version number, Allowing the End User to Specify Version Sorting, User Control of Output, End User to Modify the Listing, Suppressing Certain Columns, Showing Forbidden Files, directories aliases

References:

1. Rich Bowen, Ken Coar, “Apache Cookbook”, O’Reilly
2. Ed Sawicki , “Guide to Apache” , Cengage
3. Ivan Ristic, “Apache Security”, O’Reilly
4. Steve Silva, “Web Server Administration”, Cengage
5. Rosemary Scoular, “Apache: The Definitive Guide”, O’reilly

Accomplishments of the student after completing the course:

After completion of this course student will be able to

- Manage web server and use it effectively
- Apply security settings to web server

Course Name: Web Data Management

Course Code: MCSDE422 (1)

Objectives:

The aim of this course is to

- Articulate the main concepts, key technologies and the possible applications for web data management
- Implement business solutions over cross platform computing
- Harness the searching & indexing technologies for data management
- Understand the role of web analytics for Internet traffic

Pre-requisites:

Basic Knowledge of Operating Systems, Knowledge of Object-Oriented Programming Language, Knowledge of Database Systems

Contents:

1. **Modeling Web Data**
   XML, Web Data Management with XML, XML dialects and standards, XPath and XQuery Basics, Working with XPath, FLWOR expressions in XQuery, XPath Foundations, Typing, Typing, Automata on Words & Trees, Schema Languages for XML, Typing Graph Data, XML Query Evaluation, Managing an XML Database, Tree Pattern Evaluation using SAX

2. **Web Data Semantics and Integration**
   Ontologies, RDF and OWL, Querying Data through Ontologies, Data Integration, Wrappers and Data Extraction, Ontologies in Practice, Mashups with Yahoo! Pipes and XProc

3. **Building Web Scale Applications**
   Web Search, Parsing the Web, Web Information Retrieval, Overview of Distributed Systems, Distributed Access Structures, Distributed Computing, Full-Text Indexing, Using Recommendations, Large-Scale Data Management, Using Semi-Structured Database

4. **Web Analytics**
   Why Understanding Your Web Traffic is Important to Your Business - Available Methodologies and Their Accuracy , Page Tags and Log Files, Apply key concepts of Web Analytics, Introduction to Web Analytics Apply key tools and diagnostics associated with Web analytics, Google Analytics Features, Benefits and Limitations , Open Source Tool , Tracking the Mobile Visitor, Use free open source Web analytics tools to collect, identify information and data Prepare Embedded JavaScript Page Tracking Code, Reports Explained , Web Analytics, Up and Running , Traffic Sources: AdWords, Implement website traffic reports

References:

1. Abiteboul S., et. Al., “Web Data Management”, Cambridge University Press (Online: http://webdam.inria.fr/Jorge/files/wdm.pdf)
2. Brian Clifton, Advance Web Metrics with Google Analytics, Sybex
3. Avinash Kaushik., “Web Analytics 2.0, The Art of Online Accountability and Science of Customer” , Wiley
4. Michael Beasley, “Practical Web Analytics for User Experience”, Morgan Kaufmann
5. Stephan A. Miller, Piwik “Web Analytics Essentials”, Packt
6. Jim Sterne , “Web Metrics: Proven Methods for Measuring Web Site Success”, Wiley

Accomplishments of the student after completing the Course:
At the end of the course, students will be able to

- Decide and implement appropriate data model for WWW
- Design customized search engines
- Perform Web Analytics

Course Name: Machine Learning

Course Code: MCSDE421 (2)

Objectives:

The main objective of this course is to enable students to

- Develop a strong theoretical foundation for understanding of state of art Machine Learning Algorithms
- Identify, formulate and solve machine learning problems that arise in practical applications
- Explore various tools and technologies that enable Machine Learning

Prerequisites:

Mathematical Foundations, Programming in C++/Python

Contents:

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

2. **Linear Regression & Logistic Regression**
   Scatter diagram, Model representation for single variable, Single variable Cost Function, Least Square line fit, Normal Equations, Gradient Descent method for Linear Regression, Assumptions in linear regression, properties of regression line, Model Performance through R^2, Multivariable model representation, Multivariable cost function, multiple linear regression, Normal Equations and non-invertibility , Gradient Descent method for multiple linear regression , Overfitting, Underfitting, Bias and variance , Regularization, Issues of using Linear Regression in Classification, Sigmoid function, odds of an event, Logit function, Decision Boundary, Maximum Likelihood function, Linear regression verses Logistic Regression, Cost function, Multi-classification , confusion matrix, statistical measures to measure binary classification : Recall, sensitivity, specificity, precision, accuracy, pros and cons of logistic regression

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

4. **Unsupervised learning**
   Cluster Analysis, Classification and Clustering , Definition of Clusters ,Clustering Applications , Distance measures, Proximity Measures for Discrete Variables, Proximity Measures for Mixed Variables, Partitional Clustering, Clustering Criteria, K-Means

Algorithm, Fuzzy Clustering , Hierarchical Clustering, Agglomerative Hierarchical Clustering, Divisive Hierarchical Clustering, Cluster Validity, External Criteria, Internal Criteria

References:

1. Mitchell T., “Machine Learning”, McGraw Hill Publications
2. Grus J,. “Data Science from Scratch”, O’Reilly Publications
3. Alpaydin E., “Introduction to Machine Learning”, MIT Press
4. Coelho R., “Building Machine Learning Systems with Python”, Packt Publishing Ltd.
5. Marsland S., “MACHINE LEARNING: An Algorithmic Perspective, CRC Press
6. Wunsch D,. Xu R,. “Clustering”, IEEE Press
7. “Introduction to Machine Learning”; Ethem Alpaydın; The MIT Press

Accomplishments of the student after completing the course:

After completion of the course, students should be able to

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

----

Course Name: Deep Learning

Course Code: MCSDE422 (2)

Objectives:

The aim of this course is to enable students to

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

Prerequisites:

Basics of Linear Algebra, Calculus, Probability and Machine Learning Fundamentals

Contents:

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

2. **Convolutional Networks, Recurrent and Recursive Nets**
   Concept of Convolution, Convolution Operation, Pooling, Stride, Convolution Modules, Efficient Convolution Algorithms, Unfolding Flow graphs and Parameter Sharing, Recurrent Neural Networks, Bidirectional RNNs, Deep Recurrent Architecture, Recursive Neural Networks, Auto-Regressive Networks, Recurrent Vs. Recursive Neural Nets

3. **Regularization of Deep Models**
   Importance of Regularization, L1 and L2 Regularization, Parameter Norm Penalty, Regularization as Constrained Optimization, Weight Regularization, Classical Regularization as Noise Robustness, Dropout, Drop Connect, Multi-Task Learning, Adversarial Training

4. **Optimization for Training Deep Models & Implementations**
   Need of Optimization for Model Training, Challenges in Optimization, Basic Algorithms, Adaptive Learning, Second order methods, Natural gradient methods, Global Optimization, Image Classification, Types of Image Classifiers, Building a deep neural architecture, Building feature set, Preparing Data for training, Adding Dropout, Understanding Alex Net and Google LeNet, Using existing architectures in applications, Using inbuilt Tensor Flow functionality to build a Convolutional Neural Network

References:

1. Ian Goodfellow, Yoshua Bengio and Aaron Courville, “Deep Learning”, MIT Press

2. Jeff Heaton, "Artificial Intelligence for Humans: Deep Learning and Neural Networks Book – 3", Heaton Research Inc.
3. Giancarlo Zaccone, "Deep Learning with Tensor Flow", Packt Publishing Ltd.

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Understand various algorithms for deep learning
- Understand the concepts and architecture related to deep learning
- Design models to solve real life challenges using deep learning

----

Course Name: Blockchain Technology

Course Code: MCSDE421 (3)

Objectives:

The aim of this course is

- To introduce block chain technology and Cryptocurrency to the students
- To make students to integrate ideas from blockchain technology into their own projects

Prerequisites:

Distributed Databases

Contents:

1. **Blockchain Fundamentals & Cryptocurrency**
   Overview of Cryptography, Introduction to Blockchain, Components, Concept of Block, Block Chain Types and Consensus Mechanism, Cryptocurrency Basics, Types of Cryptocurrency, Cryptocurrency Usage : Ecosystem Players, Cryptomining, Airdrop, Token and Coin Burning, Investing and Trading, Cryptocurrency Safety

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

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

4. **Security & Applications of Blockchain**
   Security Aspect in Bitcoin, Challenges, Performance and Scalability, Identity Management and Authentication, Regulatory Compliance and Assurance, Safeguarding Blockchain Smart Contract, Uses of Blockchain in Computing, Limitations and Challenges in Blockchain, Blockchain Platform using Python

References:

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

Accomplishments of the student after completing the Course:

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

- Understand Blockchain Mechanism
- Understand Various Applications of Blockchain

Course Name: Internet of Things

Course Code: MCSDE422(3)

Objectives:

The aim of this course is to enable students to

- Understand general concepts of Internet of Things (IoT)
- Recognize various devices, sensors and applications
- Apply design concept to IoT solutions
- Analyze various M2M and IoT architectures
- Evaluate design issues in IoT applications
- Create IoT solutions using sensors, actuators and Devices

Prerequisites:

Basics of Wireless networking, Network Security

Contents:

1. **Introduction to IoT**
   Sensing, Actuation, Networking basics, Communication Protocols, Sensor Networks, Machine-to-Machine Communications, IoT Definition, Characteristics. IoT Functional Blocks, Physical design of IoT, Logical design of IoT, Communication Protocols, IoT enabling technologies, Sensor Networks, Machine-to-Machine Communications

2. **IoT & M2M**
   Difference between IoT and M2M, IoT architecture, Software define Network, SDN for IoT, Data Handling and Analytics, Cloud Computing, Sensor-Cloud, Fog Computing

3. **IoT Architecture**
   Introduction, State of the art, Architecture Reference Model- Introduction, Reference Model and architecture, IoT reference Model, IoT Reference Architecture- Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views

4. **IOT Applications and Challenges**
   Case studies: Lighting as a service, Intelligent Traffic systems, Smart Parking, Smart water management, IoT for smart cities, *IoT in Indian Scenario*: IoT and Aadhaar IoT for health services. IoT for financial inclusion IoT for rural empowerment

   Challenges in IoT implementation: Big Data Management, Connectivity challenges, Mission critical applications, security and privacy issues

References:

1. Vijay Madisetti and Arshdeep Bahga, “*Internet of Things (A Hands-on-Approach)*” ,1st edition, Orient Blackswan, Universities Press
2. Francis daCosta, “*Rethinking the Internet of Things: A Scalable Approach to Connecting everything*”, 1st edition, Apress Publications

3. Cuno Pfister, "*Getting Started with the Internet of Things*", O'Reilly Media, 2011, ISBN: 978-1-4493-357-1

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Apply IoT concepts and IoT Standards
- Understand Components and relevance of IoT System for the future
- Build IoT Applications
- Apply IoT in smart city environment in Indian scenario
- Analyze challenges in IoT implementation

Subject Name: Web Server Management - Practical

Subject Code: MCSDE423(1)

Objectives:

To enable students to know

- Install & Configure Web Server
- Setup Web Server for Security
- Optimize Web Server for better Performance

Contents:

1. **Setting up Apache Web Server**

   Install Apache on Linux and Windows, Debian Packages, Subversion Sources, Apache Toolbox, Starting and stopping Apache, Restarting Apache, Apache uninstallation, Finding Apache's files location Files

   Installing a Generic installation modules, third-party modules, Installing Unix mod_dav, Installing Windows mod_dav, Installing installation mod_perl, Unix Unix mod_perl, Installing Unix mod_php, Installing Windows mod_php ,Installing installation mod_ssl, finding Modules Using modules locating, modules.apache.org, Installing mod_security,

   Setting Up Name-Based Virtual Hosts, default virtual hosts, address-based virtual hosts, virtual hosts address-based mixing with name-based, Mass Virtual Hosting with mod_vhost_alias, Mass Virtual Hosting Using Rewrite Rules, Logging for virtual hosts, Logfile, port-based virtual hosts, Displaying the Same Content on Several Addresses, Virtual Hosts in a Database

   Dealing with Logs, Syslog and User Directories, Using CGI, Dealing with URLs

2. **Applying Security, Dealing with Proxies & Dynamic Content:**

   Using .htpasswd Files, Digest Authentication, Security in a Subdirectory, Lifting Restrictions, Selectively, files ownership, MySQL databases User Credentials, authentication usernames, authentication passwords, brute-force password attacks, Digest authentication, Credentials Embedded in URLs, WebDAV security, WebDAV enabling, proxies URL access, files wrappers, module sets, Web Root, Restricting Range Requests, DoS attacks, mod_security, Chrooting Apache with mod_security, authentication migrating, Mixing read-only access, Using redirecting permanent

   Installing SSL, Windows SSL installation, SSL Certificates , SSL (Secure Socket Layers) CAs, Serving a SSL, Authenticating with Client authentication SSL, virtual hosts SSL, SSL certificates wild card

   Securing Your Proxy Server, Open Mail Relay, Forwarding Requests to Another Server, Blocking Proxied Requests, mod_perl proxying content, caching proxy servers, Filtering Proxied Content, authentication Proxied Server, proxies, mod_proxy_balancer, proxies virtual hosts, FTP

Enabling a CGI Directory, CGI scripts enabling, CGI directories default documents, Using CGI programs launching, Using CGI scripts extensions, CGI testing setup, CGI form parameters, SSIs (Server-Side Includes) , Displaying date, last modified date, Including a Standard Header, Including the CGI program output, CGI scripts, CPAN, mod_perl Handler, PHP script handling, PHP installation, CGI parsing output, Parsing SSIs, Getting Perl scripts, Enabling Python Script Handling

References:

1) Rich Bowen, Ken Coar, “Apache Cookbook”, O’Reilly
2) Ed Sawicki , “Guide to Apache” , Cengage
3) Ivan Ristic, “Apache Security”, O’Reilly
4) Steve Silva, “Web Server Administration”, Cengage
5) Rosemary Scoular, “Apache: The Definitive Guide”, Oreilly

Accomplishments of the student after completing the course:

After completion of this course student will be able to know

- To manage web server and use it effectively.

------ ---------

Course Name: Web Data Management - Practical

Course Code: MCSDE424(1)

Objectives:

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

- Implement Web Data Modeling & Integration of Web Data
- Implement business solutions over cross platform computing
- Implement the searching & indexing technologies for data management
- Using web analytics for Internet traffic

Pre-requisites:

Basic Knowledge of Operating Systems, Knowledge of Object-Oriented Programming Language, Knowledge of Database Systems

Contents:

1. **Modeling Web Data & Web Data Semantics Integration**

   EXIST: Installing EXIST, Getting started with EXIST, Running XPath and XQuery queries with the sandbox, Programming with EXIST, Projects.

   Tree Pattern Evaluation using SAX

   Wrappers & Data Extraction with XSLT, Working with YAGO, Mashups with Yahoo! Pipes and XProc

2. **Web Scale Applications**

   Overview of LUCENE, Installing and configuring LUCENE, Indexing plain-text, Tuning the scoring

   Installing and running HADOOP, Running MAPREDUCE jobs, Running PIGLATIN scripts, Running in cluster mode

   Using COUCHDB, Dealing with data in COUCHDB, Using JSON Semi-structured data

   Using Google Analytics

References:

1. Abiteboul S., et. Al., “Web Data Management”, Cambridge University Press (Online: http://webdam.inria.fr/Jorge/files/wdm.pdf)
2. Brian Clifton, Advance Web Metrics with Google Analytics, Sybex
3. Avinash Kaushik., “Web Analytics 2.0, The Art of Online Accountability and Science of Customer” , Wiley

4. Michael Beasley, “Practical Web Analytics for User Experience”, Morgan Kaufmann
5. Stephan A. Miller, Piwik “Web Analytics Essentials”, Packt
6. Jim Sterne , “Web Metrics: Proven Methods for Measuring Web Site Success”, Wiley

Accomplishments of the student after completing the Course:

- Ability to decide and implement appropriate data model for WWW
- Ability to design customized search engines
- Ability to perform Web Analytics

Course Name: Machine Learning - Practical

Course Code: MCSDE423(2)

Objectives:

The main objective of this course is to enable students to

- Implement the theoretical knowledge of Machine Learning through programming
- Identify, formulate and solve machine learning problems that arise in practical applications
- Explore various tools and technologies that enable Machine Learning

Prerequisites:

- Mathematical Foundations, Programming in C++/Python

Contents:

1. Regression & K-Nearest Neighbor

Implementation of following concepts:

Linear Regression: Simple linear regression, Multiple linear regression, Model evaluation (R-squared, mean squared error), Regularization (Lasso, Ridge)

Logistic Regression: Sigmoid function, Binary classification using logistic regression, Multiclass classification using logistic regression, Model evaluation (confusion matrix, accuracy, precision, recall, F1score), Regularization (Lasso, Ridge)

K-Nearest Neighbors: Distance metrics (Euclidean, Manhattan, Cosine), KNN algorithm for classification and regression, Model evaluation (cross-validation)

2. Linear Regression & Logistic Regression

Implementation of following concepts:

Decision Trees: Information gain and entropy, ID3 and C4.5 algorithms for decision tree construction, Pruning decision trees, Model evaluation (confusion matrix, accuracy, precision, recall, F1-score)

Random Forests: Bagging, Construction of random forests for classification and regression, Model evaluation (out-of-bag error, feature importance)

Support Vector Machines: Maximum margin classifier, soft margin classifier, Non-linear SVM using kernel methods, Model evaluation (confusion matrix, accuracy, precision, recall, F1-score)

Unsupervised Learning: K-means clustering, Hierarchical clustering, Principal component analysis, Anomaly detection

Ensemble Methods: Boosting (AdaBoost), Bagging (Random Forest), Stacking

References:

1. Mitchell T., “Machine Learning”, McGraw Hill Publications
2. Grus J,. “Data Science from Scratch”, O’Reilly Publications
3. Alpaydin E., “Introduction to Machine Learning”, MIT Press
4. Coelho R., “Building Machine Learning Systems with Python”, Packt Publishing Ltd.
5. Marsland S., “MACHINE LEARNING: An Algorithmic Perspective, CRC Press
6. Wunsch D,. Xu R,. “Clustering”, IEEE Press
7. “Introduction to Machine Learning”; Ethem Alpaydın; The MIT Press

Accomplishments of the student after completing the course:

After completion of the course, students should be able to:

- Implement various machine learning algorithms
- Understand how to evaluate models generated from data
- Develop application involving computer vision and Natural Language Processing

***** *****

Course Name: Deep Learning - Practical

Course Code: MCSDE424(2)

Objectives:

The aim of this course is to enable students to

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

Prerequisites:

Basics of Linear Algebra, Calculus, Probability and Machine Learning Fundamentals

Contents:

1. **Deep Learning Frameworks**
   Overview & Installation of deep learning frameworks like TensorFlow, Keras, PyTorch
   Implementing Artificial neural networks (ANN), Activation functions (sigmoid, ReLU, tanh), Gradient descent optimization, Backpropagation algorithm

2. **Convolutional Neural Networks, Recurrent and Recursive Nets**
   Convolutional Neural Networks (CNN)
   Convolution operation, Pooling layer, Image classification using CNN
   Recurrent Neural Networks (RNN)
   Sequence data and LSTM networks, Text classification using RNN, Language generation using RNN

   Using pre-trained models for new tasks, Fine-tuning pre-trained models, Transfer learning for computer vision tasks

References:

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

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Understand various algorithms for deep learning
- Understand the concepts and architecture related to deep learning
- Design models to solve real life challenges using deep learning

----

Course Name: Blockchain Technology - Practical

Course Code: MCSDE423(3)

Objectives:

The aim of this course is

- To implement blockchain technology and Cryptocurrency using programming
- Students can integrate ideas from blockchain technology into their own projects.

Prerequisites: Database Management System

Contents:

1. **Working with Ethereum**
   Ethereum's Four Stages of Development, Ethereum: A General-Purpose Blockchain Ethereum's Components, From General-Purpose Blockchains to Decentralized Applications, DApps)

   Ethereum Basics
   Ether Currency Units, Choosing an Ethereum Wallet, Control and Responsibility, Getting Started with MetaMask, Creating a Wallet, Switching Networks, Getting Some Test Ether, Sending Ether from MetaMask, Exploring the Transaction History of an Address, Introducing the World Computer, Externally Owned Accounts (EOAs) and Contracts
   A Simple Contract: A Test Ether Faucet, Compiling the Faucet Contract, Creating the Contract on the Blockchain, Interacting with the Contract, Viewing the Contract Address in a Block Explorer, Funding the Contract, Withdrawing from Our Contract

   Ethereum Clients
   Ethereum Networks, Running a full Node, Full Node Advantages and Disadvantages, Public Testnet Advantages and Disadvantages, Local Blockchain Simulation Advantages and Disadvantages, Running an Ethereum Client, Hardware Requirements for a Full Node, Software Requirements for Building and Running a Client (Node), Parity, Go-Ethereum (Geth), The First Synchronization of Ethereum-Based Blockchains, Running Geth or Parity, The JSON-RPC Interface, Remote Ethereum Clients, Mobile (Smartphone) Wallets, Browser Wallets, Ethereum's Cryptographic Hash Function, Ethereum Addresses, Ethereum Address Formats, Inter Exchange Client Address Protocol, Hex Encoding with Checksum in Capitalization (EIP-55)

2. **Wallets, Transactions & Smart Contracts:**
   Wallet Technology Overview, Nondeterministic (Random) Wallets, Deterministic (Seeded) Wallets, Hierarchical Deterministic Wallets (BIP-32/BIP-44), Seeds and Mnemonic Codes (BIP-39), Wallet Best Practices, Mnemonic Code Words (BIP-39), Creating an HD Wallet from the Seed, HD Wallets (BIP-32) and Paths (BIP-43/44)

The Structure of a Transaction, The Transaction Nonce, Keeping Track of Nonces, Gaps in Nonces, Duplicate Nonces, and Confirmation, Concurrency, Transaction Origination, and Nonces, Transaction Gas, Transaction Recipient, Transaction Value and Data, Transmitting Value to EOAs and Contracts, Transmitting a Data Payload to an EOA or Contract, Special Transaction: Contract Creation, Digital Signatures, The Elliptic Curve Digital Signature Algorithm, How Digital Signatures Work, Verifying the Signature ECDSA Math, Transaction Signing in Practice, Raw Transaction Creation and Signing Raw Transaction Creation with EIP-155, The Signature Prefix Value (v) and Public Key Recovery, Separating Signing and Transmission (Offline Signing), Transaction Propagation, Recording on the Blockchain, Multiple-Signature (Multisig) Transactions

Overview of Smart Contracts, Solidity, Vyper, Tokens, Oracles, Building Decentralized Apps with Ethereum

References:

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

Accomplishments of the student after completing the Course:

- Students should be able to understand Blockchain Mechanism.
- Students should be able to understand Various Applications of Blockchain.

Course Name: Internet of Things - Practical

Course Code: MCSDE424(3)

Objectives:

The aim of this course is to enable students to

- Understand general concepts of Internet of Things (IoT)
- Recognize various devices, sensors and applications
- Apply design concept to IoT solutions
- Analyze various M2M and IoT architectures
- Evaluate design issues in IoT applications
- Create IoT solutions using sensors, actuators and Devices

Prerequisites:

Basics of Wireless networking, Network Security

Contents:

1. **Programming the IoT**
   Setting Up Your Development and Test Environment, Testing the Constrained Device App, Testing the Gateway Device App, Design Concepts, Add System Performance Tasks to the Constrained Device Application, Add System Performance Tasks to the Gateway Device Application

   Simulating Sensors and Actuators, Generating Simulated Data Using a Sensor Data Generator Class, Integrating Sensing and Actuation Simulation Within Your Application Design, Representing Sensor and Actuator Data Within the Application, Create Data Containers to Support Data Collection and Actuation, Simulating Sensors
   Simulating Actuators, Connecting Simulated Sensors with the Sensor Adapter Manager
   Connecting Simulated Actuators with the Actuator Adapter Manager, Create and Integrate the Device Data Manager, Hysteresis Management

   Emulating Sensors and Actuators, Setting Up and Configuring an Emulator, The Sense-Emu Sense HAT Emulator, Integrating Sensing and Actuation Emulation Within Your Application Design, Emulating Sensors, Emulating Actuators, Connecting Emulated Sensors with the Sensor Adapter Manager, Connecting Emulated Actuators with the Actuator Adapter Manager, Threshold Management

   Data Translation and Management Concepts, Data Translation in the Constrained Device App, Data Translation in the Gateway Device App, Proactive Disk Utilization Management

2. **IoT Integration to Objects**

   MQTT Integration, About MQTT, Connecting to a Broker, Message Passing, Control Packets and the Structure of an MQTT Message, Adding MQTT to Your Applications

Installing and Configuring an MQTT Broker, Create the MQTT Connector Abstraction Module, Add Callbacks to Support MQTT Events, Add Publish, Subscribe, and Unsubscribe Functionality, Integrate the MQTT Connector into Your CDA, Create the MQTT Connector Abstraction Module, Add Callbacks to Support MQTT Events, Add Publish, Subscribe, and Unsubscribe Functionality, Integrate the MQTT Connector into Your GDA, Security and Overall System Performance, Subscriber Callbacks, CDA to GDA Integration

CoAP Server Implementation, About CoAP, Client to Server Connections, Request Methods, Message Passing, Datagram Packets and the Structure of a CoAP Message, Add CoAP Server Functionality to the Gateway Device Application, Add CoAP Server Functionality to the Constrained Device Application, Add More Resource Handlers, Add a Custom Discovery Service, Add Dynamic Resource Creation

CoAP Client Integration, Add CoAP Client Functionality to the Constrained Device Application, Add CoAP Client Functionality to the Gateway Device App, Add a Robust OBSERVE Cancel Feature, Add Support for DELETE and POST

Edge Integration, Adding TLS Support to MQTT Broker, Add Security Features to Gateway Device App MQTT Client Connector, Add Security Features to Constrained Device App MQTT Client Connector, Adding Business Logic to the Gateway Device App, Adding Business Logic to the Constrained Device App, Add DTLS Support to Your CoAP Client and Server, Integrating IoT with Cloud Services

References:

1. Kind A., “Programming the Internet of Things”, O’Reilly Media
2. Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)” ,1st edition, Orient Blackswan, Universities Press
3. Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to Connecting everything”, 1st edition, Apress Publications
4. Cuno Pfister, “Getting Started with the Internet of Things”, O’Reilly Media, 2011, ISBN: 978-1-4493-357-1

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Apply IoT concepts and IoT Standards
- Understand Components and relevance of IoT System for the future
- Build IoT Applications
- Apply IoT in smart city environment in Indian scenario
- Analyze challenges in IoT implementation

----

5 Years Integrated MSc (Computer Science)

[Course Initiated in 2021-2023]

Semester - IX

Scheme & Course Structure

Applicable from July 2023

Course Structure with Credits

	Course Type	Subject Names & Credits	Total Credits
Semester-IX	Core Courses	• MCSCC511-Design & Analysis of Algorithms (4) • MCSCC512-Minor Project (8)	12
	Ability Enhancement Courses		----
	Skill Enhancement Courses		---
	Discipline Specific Elective Courses	• MCSDE511-Elective-5 (4) • MCSDE512-Elective-6 (4) • MCSDE513-Elective- 5– Practical (2) • MCSDE514-Elective- 6 – Practical (2)	12
	Generic Elective Courses		----
	Semester-I (Total Credits)		24

TEACHING & EXAMINATION SCHEME FOR 5 YEAR INTEGRATED M. Sc. (COMPUTER SCIENCE) COURSE

Semester-IX	Course		Hours per week								Credits
	No.	Name	Theory	Practical	Internal	Total	Lectures	Others (Tutorials)	Practical	Total	
Semester-IX	MCSCC511	Design and Analysis of Algorithms	70	***	30	100	3	1	***	4	4
	MCSCC512	Minor Project	***	140	60	200	***	2	12	14	8
	MCSDE511	Elective-5	70	***	30	100	3	1	***	4	4
	MCSDE512	Elective-6	70	***	30	100	3	1	***	4	4
	MCSDE513	Elective-5 - Practical	***	35	15	50	***	***	4	4	2
	MCSDE514	Elective-6 – Practical	***	35	15	50	***	***	4	4	2
		Total	210	210	180	600	9	5	20	34	24

Theory/Tutorial Credits → 1 Hour = 1 Credit, Practical Credits → 2 Hours = 1 Credit

Track wise subjects for Elective-5 and Elective-6			
Track	**Web Technologies**	**Artificial Intelligence & Machine Learning**	**Security**
Subject Code	MCSDE511(1)	MCSDE511(2)	MCSDE511(3)
Subject Name	Web Security	Natural Language Processing	Web Security
Subject Code	MCSDE513(1)	MCSDE513(2)	MCSDE513(3)
Subject Name	Web Security - Practical	Natural Language Processing - Practical	Web Security - Practical
Subject Code	MCSDE512(1)	MCSDE512(2)	MCSDE512(3)
Subject Name	Microservices	Computer Vision	Software Defined Networking
Subject Code	MCSDE514(1)	MCSDE514(2)	MCSDE514(3)
Subject Name	Microservices - Practical	Computer Vision - Practical	Software Defined Networking - Practical

Course Name: Design and Analysis of Algorithms

Course Code: MCSCC511

Objectives:

The aim of this course is to enable students to

- Develop proficiency in problem solving and programming.
- Carry out the Analysis of various Algorithms for mainly Time and Space Complexity.
- Get a good understanding of applications of Data Structures.
- Develop a base for advanced study in Computer Science.

Prerequisites:

Any programming language like C or C++

Contents:

1. **Introduction and familiarization:**
   The role of Algorithms in Computing, Algorithms as a technology, Asymptotic notations, Designing and Analysis of Algorithms, Insertion Sort, Divide and conquer paradigm, Merge Sort

2. **Divide and Conquer Technique, Heaps & Trees**
   The substitution method for solving recurrences, The recursion tree method for solving recurrences, The master method for solving recurrences, Heaps, Maintaining the heap property, Building a heap, The heapsort algorithm, Priority queues, Binary Search Trees & various operations, *Red-Black Trees:* Properties of red – black trees, Rotations, Insertion, Deletion

3. **Dynamic Programming & Greedy Technique**
   Elements of Dynamic Programming, Matrix-chain multiplication, Longest common subsequences, Elements of Greedy strategy, Greedy verses dynamic programming, An activity selection problem, Elements of greedy strategy, Huffman codes

4. **Elementary Graph Algorithms, String Matching, NP-Completeness**
   Representation of Graphs, Breadth-first search, Depth-first search, *Single –Source Shortest Paths:* The Bellman-Ford algorithm, Single-source shortest paths in directed acyclic graphs, Dijkstra’s algorithm, *String Matching:* The naïve string matching algorithm, The Rabin Karp algorithm, *NP-Completeness and the P & NP Classes:* Introduction, Polynomial Time & Verification, NP-Completeness and Reducibility, The Traveling Salesman Problem

References:

1. Cormen, Leiserson, Rivest, Stein, "Introduction to Algorithm", PHI
2. Parag Dave & Himanshu Dave, "Design and Analysis of Algorithms", Pearson Education
3. Michel Goodrich, Roberto Tamassia, “Algorithm design-foundation, analysis & internet examples”, Wiley
4. A V Aho, J E Hopcroft, J D Ullman, "Design and Analysis of Algorithms", Addison-Wesley Publishing
5. Brassard G. and Bratley P., Algorithms, Theory and Practices, PHI
6. Berman Kenneth, Paul Jerome, “Fundamentals of Sequential and Parallel Algorithms”, Cengage Learning
7. Anany V. Levitin ”Introduction to the Design and Analysis of Algorithms” Pearson Education publication
8. Knuth D.E., "The Art of Computer Programming", Vol. 1,2 and 3, Addison- Wesley Publishing
9. Horowitz E, Sahni S, Rajasekaran S., “Fundamentals of Computer Algorithms”, University Press

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

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

----

Course Name: Minor Project

Course Code: MCSCC512

Objectives:

The aim of this course is to

- Introduce structured development of software systems
- Acquaint students to various techniques of requirements determination
- Introduce the concepts of analysis and design
- Model the system with various software diagrams
- Develop a system using software engineering concepts
- Prepare document/report of the system

Prerequisites:

Knowledge of Programming

Contents:

1. **Introduction to System Analysis & Design**
   Concepts of System Analysis & Design, Components of a System, Users of a System, System Development Life Cycle, Role of System Analyst, Various approaches to develop a system, Using a "systems" approach to build whole system, Applying concepts of Distributed Systems Development, Apply the concepts of security while designing the Project, Adhere to OWASP rules while designing the system

2. **Structured Systems Development Approach**
   Requirements Gathering, surveying existing systems, Performing Feasibility Study before building a system, Formulating SRS, Designing Data Flow Diagrams, Entity-Relationship Diagrams, Designing Data Dictionary, Designing effective Input and Reports, Implementing EOD, EOM and EOY Procedures, Documenting the system

Guidelines for Project:

1. Projects can be created by a single student or in a group of 2-3 students
2. The definition of project is to be submitted within 15 days of the starting of the semester
3. The project should be free from plagiarism
4. A student/group is required to report to their project guides on regular basis
5. A student/group is required to present the project during internal exams
6. It is advisable that the project must include different forms and reports
7. It is necessary to prepare system diagrams, data dictionary as per the convention
8. Project documentation should be prepared for the semester end examination

9. In the semester end examination, the project should be presented to the examiners along with the documentation(report)

References:

1. Kendall & Kendall. “System Analysis and Design ”, Prentice Hall India (PHI)
2. James Sen, “ Analysis and Design of Information System”, PHI
3. Arthur Langer, “ Analysis and Design of Information Systems “, Springer
4. Jeff Johnson, “ Designing with the Mind in Mind “, Morgan Kaufmann
5. Dennis, Wixom, Roth “System Analysis and Design”, Wiley India
6. Joseph Valacich, Joey F. George, “Modern Systems Analysis and Design”, Pearson
7. Priti Srinivas Sajja., “Essence of System Analysis and Design”, Springer

Accomplishments of the student after completing the Course:
After completion of this course, students will be able to

- Analyse and Design a system based on user requirements
- Develop a software system using software engineering concepts
- Prepare a document for the system

Course Name: Web Security

Course Code: MCSDE511 (1)

Objectives:

The aim of this course is to

- Know necessity to secure web applications
- Understand the different risks to web applications
- Understand the steps required to mitigate those risks

Prerequisites:

Fundamentals of Client Side Scripting, Server Side Scripting & Database

Contents:

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

2. **Authentication Security & Injection Attacks**
   Authentication Techniques, Design Flaws in Authentication, Implementation Flaws in Authentication, Securing Authentication, Path Traversal Attacks, Injecting into Interpreted Contexts, SQL Injection, NoSQL Injection, XPath Injection, LDAP Injection, XML Injection, Http Injection, Mail Service Injection

3. **Cross Site Scripting (XSS) & User Attacks**
   Types of XSS, XSS in Real World, Finding and Exploiting XSS Vulnerabilities, Preventing XSS Attacks, Inducing User Actions, Capturing Cross-Domain Data, Client-Side Injection Attacks, Local Privacy Attacks, ActiveX Control attacks, Browser Attacks

4. **Source Code Analysis**
   Approaches to Code Review, Signatures of Common Vulnerabilities, Analysis of Java platform, Analysis of ASP.NET platform, Analysis of PHP, Analysis of Perl, Analysis of JavaScript, Analysis of SQL

References:

1. Dafydd Stuttard, “The Web Application Hacker’s Handbook, 2nd Edition”, Wiley India Pvt. Ltd.
2. Hanqing Wu, Liz Zhao, “Web Security A WhiteHat Perspective”, Taylor & Francis Ltd

3. Jonathan LeBlanc, Tim Messerschmidt, “Identity and Data Security for Web Development Best Practices ”, O'Reilly Media
4. Ron Lepofsky , “ The Manager's Guide to Web Application Security ”, Apress

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Detect and solve common web application security vulnerabilities
- Secure web applications

Course Name: Microservices

Course Code: MCSDE513 (1)

Objectives:

The aim of this course is to

- Develop proficiency in Microservices Design
- Divide large applications into self-contained units
- Reduce complexity and improve flexibility

Prerequisites:

Java Programming Language

Contents:

1. **Introduction and familiarization**
   Overview of Microservices, Key Concepts of Microservices, Independent Deployability, Modeled Around a Business Domain, Owning Their Own State, Size, Flexibility, Alignment of Architecture and Organization, The Monolith, The Single-Process Monolith, The Modular Monolith, The Distributed Monolith, Monoliths and Delivery Contention, Advantages of Monoliths, Enabling Technology, Log Aggregation and Distributed Tracing, Containers and Kubernetes, Streaming, Public Cloud and Serverless, Advantages of Microservices, Technology Heterogeneity, Robustness, Scaling, Ease of Deployment, Organizational Alignment, Composability, Microservice Pain Points, Developer Experience, Technology Overload, Cost, Reporting, Monitoring and Troubleshooting, Security, Testing, Latency, Data Consistency, Using Microservices, Modeling Microservices, Microservices Boundary, Information Hiding, Cohesion, Coupling, Types of Coupling, Domain Driven Design, Ubiquitous Language, Aggregate, Bounded Context, Mapping Aggregates and Bounded Contexts to Microservices, Event Storming, Volatility, Data, Technology, Mixing Models & Exceptions, Splitting the Monolith, Incremental Migration, The Dangers of Premature Decomposition, Decomposition by Layer, Code First, Data First, Useful Decompositional Patterns, Strangler Fig Pattern, Parallel Run, Feature Toggle, Data Decomposition Concerns, Performance, Data Integrity, Transactions, Tooling, Reporting Database, Microservice Communication Styles, From In-Process to Inter-Process, Performance, Changing Interfaces, Error Handling, Technology for Inter-Process Communication: So Many Choices, Styles of Microservice Communication, Mix and Match, Pattern: Synchronous Blocking, Advantages & Disadvantages, Pattern: Asynchronous Nonblocking, Communication Through Common Data, Request-Response Communication, Synchronous Vs Asynchronous, Event-Driven Communication

2. **Implementing Microservices**
   Implementing Microservice Communication, Looking for the Ideal Technology, Make Backward Compatibility Easy, Make Your Interface Explicit, Keep Your APIs Technology Agnostic, Make Your Service Simple for Consumers, Hide Internal Implementation Detail,

Technology Choices, Remote Procedure Calls, REST, GraphQL, Message Brokers, Serialization Formats, Textual Formats, Binary Formats, Schemas, Structural Versus Semantic Contract Breakages, Using Schemas, Handling Change Between Microservices, Avoiding Breaking Changes, Expansion Changes, Tolerant Reader, Right Technology, Explicit Interface, Catch Accidental Breaking Changes Early, Managing Breaking Changes, Lockstep Deployment, Coexist Incompatible Microservice Versions, Emulate the Old Interface, The Social Contract, Tracking Usage, Extreme Measures, Sharing Code via Libraries, Service Discovery, Domain Name System (DNS), Dynamic Service Registries, Service Meshes and API Gateways, API Gateways, Service Meshes, Other Protocols, Documenting Services, Explicit Schemas, The Self-Describing System, Workflow, Database Transactions, ACID Transactions, Distributed Transactions, Sagas, Saga Failure Modes, Implementing Sagas, Sagas Versus Distributed Transactions, Building Webservices, A Brief Introduction to Continuous Integration, Branching Models, Build Pipelines and Continuous Delivery, Tooling, Trade-Offs and Environments, Artifact Creation, Mapping Source Code and Builds to Microservices, One Giant Repo, One Giant Build, *Pattern:* One Repository per Microservice, *Pattern:* Monorepo, Deployment of Microservice, From Logical to Physical, Multiple Instances, The Database, Environments, Principles of Microservice Deployment, Isolated Execution, Focus on Automation, Infrastructure as Code (IAC), Zero-Downtime Deployment, Desired State Management, Deployment Options, Physical Machines, Virtual Machines, Containers, Application Containers, Platform as a Service (PaaS), Function as a Service (FaaS), Kubernetes and Container Orchestration, The Case for Container Orchestration, A Simplified View of Kubernetes Concepts, Multitenancy and Federation, The Cloud Native Computing Federation, Platforms and Portability, Progressive Delivery, Separating Deployment from Release, Progressive Delivery, Feature Toggles, Canary Release, Parallel Run, Testing Microservices, Types of Tests, Test Scope, Unit Tests, Service Tests, End-to-End Tests, Trade-Offs, Implementing Service Tests, Mocking or Stubbing, A Smarter Stub Service, Flaky and Brittle Tests, The Metaversion, Lack of Independent Testability, Contract Tests and Consumer-Driven Contracts (CDCs),Developer Experience, From Preproduction to In-Production Testing, Types of In-Production Testing, Making Testing in Production Safe, Mean Time to Repair over Mean Time Between Failures, Cross-Functional Testing

3. Improving Microservices

From Monitoring to Observability, Disruption, Panic, and Confusion, Single Microservice, Single Server, Single Microservice, Multiple Servers, Multiple Services, Multiple Servers, Observability Versus Monitoring, The Pillars of Observability, Building Blocks for Observability, Log Aggregation, Metrics Aggregation, Distributed Tracing, Alerting, Semantic Monitoring, Testing in Production, Standardization, Selecting Tools, Democratic, Easy to Integrate, Provide Context, Real-Time, Scaling, Security, Core Principles, Principle of Least Privilege, Defense in Depth, Automation, Build Security into the Delivery Process, The Five Functions of Cybersecurity, Foundations of Application Security, Credentials, Patching, Backups, Rebuild, Implicit Trust Versus Zero Trust, Implicit Trust, Zero Trust, Securing Data, Data in Transit, Data at Rest, Authentication and Authorization, Service-to-Service Authentication, Human Authentication, Common Single Sign-On Implementations, Single Sign-On Gateway, Fine-Grained Authorization, The Confused Deputy Problem, Centralized, Upstream Authorization, Decentralizing Authorization, JSON Web Tokens, Resiliency, Robustness, Rebound, Graceful Extensibility, Sustained Adaptability, And Microservice Architecture, Degrading Functionality, Stability Patterns, Time-Outs, Retries, Bulkheads, Circuit Breakers, Isolation, Redundancy, Middleware, Idempotency, CAP Theorem, Sacrificing Consistency, Sacrificing Availability, Sacrificing Partition Tolerance, AP or CP, Production Experiments, Scaling, The Four Axes of Scaling, Vertical Scaling, Horizontal Duplication, Data Partitioning, Functional Decomposition, Combining Models, Starting Small, Caching, For Performance, For Scale, For Robustness, Where to Cache, Invalidation, The Golden Rule of Caching, Freshness Versus Optimization, Cache Poisoning, Autoscaling

4. **People & Organization**
User Interfaces, Ownership Models, Drivers for Dedicated Frontend Teams, Toward Stream-Aligned Teams, Sharing Specialists, Ensuring Consistency, Working Through Technical Challenges, Pattern: Monolithic Frontend, Micro Frontends, Page-Based Decomposition, Widget-Based Decomposition, Implementation, Central Aggregating Gateway, Different Types of User Interfaces, Multiple Concerns, Backend for Frontend (BFF), BFFs for Desktop Web and Beyond, GraphQL, A Hybrid Approach, Organizational Structures, Loosely Coupled Organizations, Conway’s Law, Strong Versus Collective Ownership, Strong Ownership, Collective Ownership, At a Team Level Versus an Organizational Level, Balancing Models, Enabling Teams, Communities of Practice, The Platform, Shared Microservices, Cross-Cutting Changes, Delivery Bottlenecks, Internal Open Source, Role of the Core Committers, Maturity, Tooling, Pluggable, Modular Microservices, Change Reviews, The Orphaned Service, The Evolutionary Architect, Making Change Possible, An Evolutionary Vision for the Architect, Defining System Boundaries, A Social Construct, Habitability, A Principled Approach, Strategic Goals, Principles, Practices, Combining Principles and Practices, Guiding an Evolutionary Architecture, Architecture in a Stream-Aligned Organization, Building a Team, The Required Standard, Monitoring, Interfaces, Architectural Safety, Governance and the Paved Road, Exemplars, Tailored Microservice Template, Exception Handling

References:

1. Newman S., "Building Microservices”, O’Reilly Media
2. Erl Thomas, “Service Oriented Architecture”, Prentice Hall
3. Morris, Kief. “Infrastructure As Code. O’Reilly
4. Wolff E., “Microservices – Flexible Software Architecture”, Pearson Education
5. Mitra et. al., “Microservices Architecture”, O’Reilly Media
6. Carnell J., Sanchez H., “Spring Microservices in Action”, Manning Publications

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Detect and solve common web application security vulnerabilities
- Build separate self-contained units of applications.
- Select the best design in implementing microservices based architecture
- Build scalable and flexible applications

Course Name: Natural Language Processing

Course Code: MCSDE511 (2)

Objectives:

The aim of this course is to enable students to

- Study the key concepts pertaining to Linguistics and NLP that are used to describe and analyze natural language
- Gain insights into statistical and semantic approaches to NLP
- Apply basic principles of machine learning to natural language data
- Appreciate the use standard software packages for machine learning in the domain of NLP
- Understand how data structures and algorithms are used in NLP

Prerequisites:

Basic Knowledge of Introduction to AI & Machine Learning, Probability & Statistics, Linear Algebra

Contents:

1. **Introduction to natural language processing**
   Structural features of texts in natural language, ambiguity on all levels of language, the main challenges of natural language processing, *Basic approaches to problem solving*: manually written rules and machine learning
   <u>Basic text processing and edit distance</u>
   Preprocessing: tokenization and segmentation, *Normalization of words*: stemming, lemmatization, Bag of Words, TF-IDF, morphological analyzers, regular expressions, edit distance

2. **Language models, Tagging & Hidden Markov Models**
   N-grams, perplexity, *The use of language models*: input prediction, error correction, speech recognition, text generation, *Tagging:* PoS tagging, named entity recognition as a tagging problem, Hidden Markov models, their advantages and disadvantages, the Viterbi algorithm

3. **Text classification, Parsing & Computational Semantics**
   Classification problems, naive Bayes classifier, text classification, sentiment analysis
   Parsing: Constituency and dependency trees; context-free grammar; probabilistic approach to parsing; lexicalized PCFGs; CKY algorithm
   Computational Semantics: Word senses and meanings, WordNet, Semantic similarity measures: thesaurus-based and distributional methods

4. **Vector Space Models, Summarization, NLP with Sequence Models**
   Word2vec and doc2vec, Word embeddings, Character to Sentence Embeddings
   Text summarization
   Extractive and abstractive summarization, multiple-document summarization, query-based summarization, supervised and unsupervised learning, evaluation of summarization systems, ROUGE
   NLP with Sequence Models
   Recurrent Neural Networks: LSTM, GRU, Neural Networks for sentiment analysis, Deep N-Grams

References:

1. J. Perkins, "Python 3 Text Processing with NLTK 3 Cookbook", Packt Publishing Ltd, 2014
2. C. D. Manning, P. Raghavan, H. Schütze, et al., "Introduction to information retrieval: vol. 1.", Cambridge University Press Cambridge, 2008
3. Steven Bird, Ewan Klein and Edward Loper, "Natural language processing with Python: analyzing text with the natural language toolkit", O'Reilly Media, Inc., 2009
4. J. Pustejovsky and A. Stubbs, "Natural language annotation for machine learning", O'Reilly Media Inc., 2012
5. U. S. Tiwary and T. Siddiqui, "Natural language processing and information retrieval", Oxford University Press Inc., 2008
6. D. Chopra, N. Joshi, and I. Mathur, "Mastering Natural Language Processing with Python", Packt Publishing Ltd., 2016

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

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

Course Name: Computer Vision

Course Code: MCSDE513 (2)

Objectives:

The aim of this course is to enable students to

- Learn and understand the concepts of artificial intelligence
- Gain sufficient background knowledge for plentiful challenging applications in automation

Prerequisites:

Introductory course in Linear Algebra, Calculus and Probability

Contents:

1. **Introduction & Image Transforms**
   Overview, Smoothing, Image Morphology, Flood Fill, Resize, Image Pyramids, Thresholding operation, Convolution, Gradients and Sobel Derivatives, Laplace, Canny & Hough Transforms, Remap, Stretch, Shrink, Warp, and Rotate, Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Integral Images, Distance Transform, Histogram Equalization

2. **Image Parts and Segmentation**
   Parts and Segments, Background Subtraction, Watershed Algorithm, Image Repair by Inpainting, Mean-Shift Segmentation, Delaunay Triangulation, Voronoi Tessellation

3. **Tracking and Motion**
   The Basics of Tracking, Corner Finding, Subpixel Corners, Invariant Features, Optical Flow, Mean-Shift & Camshift Tracking, Motion Templates, Estimators, Lucas-Kanade algorithm for optical flow, Multi-scale Lucas-Kanade algorithm, Comparison of Horn-Shunck and Lucas-Kanade algorithms, Applications of optical flow

4. **Camera Models and Calibration**
   Developing Camera Model, Calibration -Concept of camera calibration and basic aim of camera calibration, Motivation for camera calibration - implications for 3D reconstruction using two calibrated cameras, Un-distortion, Putting Calibration Together, Rodrigues Transform

References:

1. Sonka & Hlavac, “Image Processing: Analysis and Machine Vision”, Springer US
2. Gonzalez, “Digital Image Processing”, Pearson
3. Richard Szeliski, “Computer Vision: Algorithms and Applications”, Springer-Verlag London Limited 2011

Accomplishments of the student after completing the Course:
At the end of the work students will be able to

- Solve challenging problems using Python programming language
- Get foundation of image formation, processing and analysis
- Understand the geometric aspects of images in spatial and frequency domain
- Gain exposure to object and image recognition with feature extraction, pattern analysis and geometric modeling
- Develop practical skills necessary to build computer vision applications including mining of visual content, image rendering, camera surveillance etc.

Course Name: Web Security

Course Code: MCSDE511 (3)

Objectives:

The aim of this course is to enable students to

- Know the necessity to secure web applications
- Understand the different risks to web applications
- Understand the steps required to mitigate those risks

Prerequisites:

Fundamentals of Client Side Scripting, Server Side Scripting & Database

Contents:

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

2. **Authentication Security & Injection Attacks**
   Authentication Techniques, Design Flaws in Authentication, Implementation Flaws in Authentication, Securing Authentication, Path Traversal Attacks, Injecting into Interpreted Contexts, SQL Injection, NoSQL Injection, XPath Injection, LDAP Injection, XML Injection, Http Injection, Mail Service Injection

3. **Cross Site Scripting (XSS) & User Attacks**
   Types of XSS, XSS in Real World, Finding and Exploiting XSS Vulnerabilities, Preventing XSS Attacks, Inducing User Actions, Capturing Cross-Domain Data, Client-Side Injection Attacks, Local Privacy Attacks, ActiveX Control attacks, Browser Attacks

4. **Source Code Analysis**
   Approaches to Code Review, Signatures of Common Vulnerabilities, Analysis of Java platform, Analysis of ASP.NET platform, Analysis of PHP, Analysis of Perl, Analysis of JavaScript, Analysis of SQL

References:

1. Dafydd Stuttard, “The Web Application Hacker’s Handbook, 2nd Edition”, Wiley India Pvt. Ltd.
2. Hanqing Wu, Liz Zhao, “Web Security A WhiteHat Perspective”, Taylor & Francis Ltd

3. Jonathan LeBlanc, Tim Messerschmidt, “Identity and Data Security for Web Development Best Practices ”, O'Reilly Media
4. Ron Lepofsky , “ The Manager's Guide to Web Application Security ”, Apress

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Detect and solve common web application security vulnerabilities
- Secure web applications

Course Name: Software Defined Networking

Course Code: MCSDE513 (3)

Objectives:

The aim of this course is to enable students to

- Deploy software defined networks
- Understand emerging networking technologies for cloud infrastructure
- Obtain skills to do research in software defined networking (SDN)
- Use software programs to perform varying and complex networking tasks

Prerequisites:

Fundamentals of Computer Networks

Contents:

1. **Introduction to SDN, Architecture & Working**
   Traditional switch architecture, Autonomous and dynamic forwarding table, Technological shifts, Evolution of switches and control planes, Consideration of cost, Needs of data center and innovations, Evolution of networking technology, Forerunners of SDN, Birth of SDN technology, Open source contributions, SDN interoperability, Overview of network virtualization, Fundamental characteristics of SDN, SDN operations, SDN devices, SDN Controller, SDN applications, Alternate SDN methods

2. **OpenFlow Specification, Working with SDN**
   Overview of OpenFlow, basics of OpenFlow, Various versions of OpenFlow, OpenFlow interoperability, Limitations of OpenFlow, SDN via APIs, SDN via hypervisor-based overlays, SDN via Opening Up the Device, Network Functions Virtualization, Potential drawbacks of open SDN

3. **SDN in different environments, NFV**
   SDN in data centres, SDN in WAN, SDN in campus networks, SDN in hospitality networks, SDN in mobile networks, SDN for service providers and carrier networks, SDN in optical networks, Definition of NFV, Standards, OPNFV, Leading NFV Vendors, SDN Vs. NFV, In-line Network Functions, Overview of Emerging Protocol, Controller, and Application Models for SDN

4 **SDN Security**
SDN & NFV Security, Microsegmentation, MTD Classification, SDN-MTD Classification, Evaluation of MTD, Attack Representation, Security Policy Management in Distributed SDN Environments, Intelligent Software Defined Security

References:

1. Paul G., Chuck B., and Timothy C., "Software Defined Networks : A Comprehensive Approach", 2nd Edition, Morgan Kaufmann
2. Huang D., et. al., "Software Defined Networking and Security", CRC Press
3. Patricia M., James A, "Software Defined Networking" , CRC Press
4. Thomas N. and Kenneth G., "SDN : Software Defined Networks", O'Reilly Media.
5. Coker O. and Azodolmolky S., "Software-Defined Networking with OpenFlow", Packt publishing ltd.
6. Dijiang H., Chowdhary A., and Pisharody S., "Software-Defined Networking and Security: From Theory to Practice", CRC Press
7. Subramanian S. and Voruganti S., "Software-Defined Networking (SDN) with OpenStack", Packt publishing ltd.

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Have thorough understanding of software defined networking
- Deploy software defined networks
- Analyze and develop solutions to solve SDN related problems
- Effectively use various tools and utilities for computer networking

----

Course Name: Web Security – Practical

Course Code: MCSDE512 (1)

Objectives:

The aim of this course is to enable students to

- Know the necessity to secure web applications
- Understand the different risks to web applications
- Understand the steps required to mitigate those risks

Prerequisites:

Fundamentals of Client Side Scripting, Server Side Scripting & Database

Contents:

1. **Setting Up Penetration Test**
   Configuring Burp Suite, Configuring the Client and Setting up the Mobile Devices, Executing an Application Penetration Test, Exploring the Stages of an Application Penetration Test, Harnessing the features of Burp Suite, Preparing for an Application Penetration Test, Identifying Vulnerabilities using Burp Suite, Detecting SQL Injection flaws, Detecting OS Command injection, Detecting XSS Vulnerabilities, Detecting XML related issues, Detecting SSTI, Detecting SSRF, Detecting CSRF, Detecting Insecure Direct Object References, Detecting security misconfigurations, Detecting insecure deserialization, Detecting OAuth-related issues, Detecting broken authentication

2. **Performing Penetration Test**
   Data exfiltration via a blind Boolean-based SQL injection, Executing OS commands using an SQL injection, Executing an out-of-band command injection, Stealing session credentials using XSS, Taking control of the user's browser using XSS, Extracting server files using XXE vulnerabilities, Performing out-of-data extraction using XXE and Burp Suite collaborator, Exploiting SSTI vulnerabilities to execute server commands, Using SSRF/XSPA to perform internal port scans, Using SSRF/XSPA to extract data from internal machines, Extracting data using Insecure Direct Object Reference (IDOR) flaws, Exploiting security misconfigurations, Using insecure deserialization to execute OS commands, Brute forcing HTTP basic authentication, Brute forcing forms, Bypassing file upload restrictions, Writing Burp suite extensions

References:

1. Lozano C., et. al., “Hands-On Application Penetration Testing with Burp Suite”, Packt Publishing

2. Baloch R., “Ethical Hacking & Penetration Testing Guide”, CRC Press
3. Khawaja G., “Practical Web Penetration Testing”, Packt Publishing
4. Dafydd Stuttard, “The Web Application Hacker’s Handbook, 2nd Edition”, Wiley India Pvt. Ltd.
5. Hanqing Wu, Liz Zhao, “Web Security A WhiteHat Perspective”, Taylor & Francis Ltd
6. Jonathan LeBlanc, Tim Messerschmidt, “Identity and Data Security for Web Development Best Practices ”, O'Reilly Media
7. Ron Lepofsky , “ The Manager's Guide to Web Application Security ”, Apress

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Detect and solve common web application security vulnerabilities
- Secure web applications

Course Name: Microservices – Practical

Course Code: MCSDE514 (1)

Objectives:

The aim of this course is to

- Develop proficiency in Microservices Design
- Divide large applications into self-contained units
- Reduce complexity and improve flexibility

Prerequisites:

Java Programming Language, Cloud Computing

Contents:

1. **Introduction and familiarization**
   Overview of Spring, Features and Attributes of Spring, Configurations with Spring, Building Microservices with Spring Boot and Java, Designing Microservice Architecture

2. **Implementing Microservices**
   Controlling your configuration with the Spring Cloud Configuration Server, Service discovery, client-side resiliency patterns, Service routing with Spring Cloud Gateway, Securing microservices, Event-driven architecture with Spring Cloud Stream, Distributed tracing with Spring Cloud Sleuth and Zipkin, Deploying Microservices

References:

1. Newman S., "Building Microservices", O'Reilly Media
2. Varanasi B., Bartkov M., "Spring REST", Apress
3. Carnell J., Sanchez H., "Spring Microservices in Action", Manning Publications
4. Erl Thomas, "Service Oriented Architecture", Prentice Hall
5. Morris, Kief. "Infrastructure As Code. O'Reilly
6. Wolff E., "Microservices – Flexible Software Architecture", Pearson Education
7. Mitra et. al., "Microservices Architecture", O'Reilly Media

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Build separate self-contained units of applications.
- Select the best design in implementing microservices based architecture
- Build scalable and flexible applications

Course Name: Natural Language Processing - Practical

Course Code: MCSDE512(2)

Objectives:

The aim of this course is to enable students to

- To study the key concepts pertaining to Linguistics and NLP that are used to describe and analyze natural language
- To gain insights into statistical and semantic approaches to NLP
- To apply basic principles of machine learning to natural language data
- Appreciate the use standard software packages for machine learning in the domain of NLP
- To understand how data structures and algorithms are used in NLP

Prerequisites:

Basic Knowledge of Introduction to AI & Machine Learning, Probability & Statistics, Linear Algebra

Contents:

1. **NLP Libraries & Text Processing**
   Overview of NLP libraries and frameworks (NLTK, spaCy, TensorFlow NLP),
   Text Preprocessing: Tokenization, Stopword removal, Stemming and Lemmatization, Text normalization
   Text Representation: Bag-of-Words model, N-grams, Term frequency-inverse document frequency (TF-IDF), Word embeddings (Word2Vec)
   Text Classification: Naive Bayes, Support vector machines (SVM), Neural networks (MLP), Deep learning models (RNN, LSTM)

2. **Sentiment Analysis, POS Tagging & Machine Translation**
   Sentiment Analysis: Lexicon-based methods, Machine learning models, Deep learning models (RNN, LSTM)
   Part-of-Speech Tagging: Hidden Markov models (HMM), Conditional random fields (CRF), Deep learning models (RNN, LSTM)
   Machine Translation: Statistical machine translation (SMT), Neural machine translation (NMT), Attention-based NMT models

References:

1. J. Perkins, "*Python 3 Text Processing with NLTK 3 Cookbook*", Packt Publishing Ltd, 2014
2. C. D. Manning, P. Raghavan, H. Schütze, et al., "*Introduction to information retrieval: vol. 1.*", Cambridge University Press Cambridge, 2008
3. Steven Bird, Ewan Klein and Edward Loper, "*Natural language processing with Python: analyzing text with the natural language toolkit*", O'Reilly Media, Inc., 2009

4. J. Pustejovsky and A. Stubbs, “*Natural language annotation for machine learning*”, O’Reilly Media Inc., 2012
5. U. S. Tiwary and T. Siddiqui, “*Natural language processing and information retrieval*”, Oxford University Press Inc., 2008
6. D. Chopra, N. Joshi, and I. Mathur, “*Mastering Natural Language Processing with Python*”, Packt Publishing Ltd., 2016

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

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

Course Name: Computer Vision - Practical

Course Code: MCSDE514(2)

Objectives:

The aim of this course is to enable students to

- Learn and understand the concepts of artificial intelligence
- Gain sufficient background knowledge for plentiful challenging applications in automation

Prerequisites:

Introductory course in Linear Algebra, Calculus and Probability

Contents:

1. **CV Libraries & Image Processing**
   Overview of CV libraries and frameworks (OpenCV, TensorFlow CV, PyTorch CV)
   Image Processing: Image representation (grayscale, RGB), Image enhancement (histogram equalization, filtering), Image transformation (rotation, scaling, translation), Erosion, Dilation, Opening, Closing

2. **Object Detection, Image Classification, Segmentation**
   Object Detection: Scale-invariant feature transform (SIFT), Speeded up robust features (SURF)
   Image Classification: K-nearest neighbors (KNN), Support vector machines (SVM), Convolutional neural networks (CNN)
   Segmentation: Thresholding, Region-based segmentation (watershed)

References:

1. Sonka & Hlavac, “Image Processing: Analysis and Machine Vision”, Springer US
2. Gonzalez, “Digital Image Processing”, Pearson
3. Richard Szeliski, “Computer Vision: Algorithms and Applications”, Springer-Verlag London Limited 2011

Accomplishments of the student after completing the Course:

At the end of the work students will be able to

- Solve challenging problems using Python programming language
- Get foundation of image formation, processing and analysis
- Understand the geometric aspects of images in spatial and frequency domain
- Gain exposure to object and image recognition with feature extraction, pattern analysis and geometric modeling

- Develop practical skills necessary to build computer vision applications including mining of visual content, image rendering, camera surveillance etc.

Course Name: Web Security – Practical

Course Code: MCSDE512 (3)

Objectives:

The aim of this course is to enable students to

- Know the necessity to secure web applications
- Understand the different risks to web applications
- Understand the steps required to mitigate those risks

Prerequisites:

Fundamentals of Client Side Scripting, Server Side Scripting & Database

Contents:

1. **Setting Up Penetration Test**
   Configuring Burp Suite, Configuring the Client and Setting up the Mobile Devices, Executing an Application Penetration Test, Exploring the Stages of an Application Penetration Test, Harnessing the features of Burp Suite, Preparing for an Application Penetration Test, Identifying Vulnerabilities using Burp Suite, Detecting SQL Injection flaws, Detecting OS Command injection, Detecting XSS Vulnerabilities, Detecting XML related issues, Detecting SSTI, Detecting SSRF, Detecting CSRF, Detecting Insecure Direct Object References, Detecting security misconfigurations, Detecting insecure deserialization, Detecting OAuth-related issues, Detecting broken authentication

2. **Performing Penetration Test**
   Data exfiltration via a blind Boolean-based SQL injection, Executing OS commands using an SQL injection, Executing an out-of-band command injection, Stealing session credentials using XSS, Taking control of the user's browser using XSS, Extracting server files using XXE vulnerabilities, Performing out-of-data extraction using XXE and Burp Suite collaborator, Exploiting SSTI vulnerabilities to execute server commands, Using SSRF/XSPA to perform internal port scans, Using SSRF/XSPA to extract data from internal machines, Extracting data using Insecure Direct Object Reference (IDOR) flaws, Exploiting security misconfigurations, Using insecure deserialization to execute OS commands, Brute forcing HTTP basic authentication, Brute forcing forms, Bypassing file upload restrictions, Writing Burp suite extensions

References:

1. Lozano C., et. al., “Hands-On Application Penetration Testing with Burp Suite”, Packt Publishing

2. Baloch R., “Ethical Hacking & Penetration Testing Guide”, CRC Press
3. Khawaja G., “Practical Web Penetration Testing”, Packt Publishing
4. Dafydd Stuttard, “The Web Application Hacker’s Handbook, 2nd Edition”, Wiley India Pvt. Ltd.
5. Hanqing Wu, Liz Zhao, “Web Security A WhiteHat Perspective”, Taylor & Francis Ltd
6. Jonathan LeBlanc, Tim Messerschmidt, “Identity and Data Security for Web Development Best Practices ”, O'Reilly Media
7. Ron Lepofsky , “ The Manager's Guide to Web Application Security ”, Apress

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Detect and solve common web application security vulnerabilities
- Secure web applications

Course Name: Software Defined Networking - Practical

Course Code: MCSDE514 (3)

Objectives:

The aim of this course is to enable students to

- Able to deploy software defined networks
- Able to create a full-fledged flow-based OpenFlow switch
- Develop own OpenFlow Controller
- Use software programs to perform varying and complex networking tasks

Prerequisites:

Fundamentals of Computer Networks

Contents:

1. **Implementation of Mininet using OpenFlow**
   Mininet Introduction, Mininet using OpenFlow, Mininet with 50x OpenFlow switches, OpenFlow Tables, SDN Architecture, SDN App Store, App install and Network flow update, REST API - SDN Application, Mininet Configuration, Mininet with Reference Controller(PTCP, Mininet with POX Controller, OpenFLow ACL, OpenFlow ACL

2. **RYU Controller Programming**
   RYU Controller programming exercises (L3 Switch, L4 Switch, Flow timeouts, Flow priorities, Flow pipelining processing, ARP Proxy, Group tables

References:

1. Paul G., Chuck B., and Timothy C., "Software Defined Networks : A Comprehensive Approach", 2nd Edition, Morgan Kaufmann
2. Huang D., et. al., "Software Defined Networking and Security", CRC Press
3. Patricia M., James A, "Software Defined Networking" , CRC Press
4. Thomas N. and Kenneth G., "SDN : Software Defined Networks", O'Reilly Media.
5. Coker O. and Azodolmolky S., "Software-Defined Networking with OpenFlow", Packt publishing ltd.
6. Dijiang H., Chowdhary A., and Pisharody S., "Software-Defined Networking and Security: From Theory to Practice", CRC Press
7. Subramanian S. and Voruganti S., "Software-Defined Networking (SDN) with OpenStack", Packt publishing ltd.

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Have thorough understanding of software defined networking
- Deploy software defined networks
- Analyze and develop solutions to solve SDN related problems
- Effectively use various tools and utilities for computer networking

----

5 Years Integrated MSc (Computer Science)

[Course Initiated in 2021-2023]

Semester - X

Scheme & Course Structure

Applicable from July 2023

Course Structure with Credits

	Course Type	Subject Names & Credits	Total Credits
Semester-X	Core Courses	• MCSCC521 Major Project	**24**
	Ability Enhancement Courses		—
	Skill Enhancement Courses		—
	Discipline Specific Elective Courses		—
	Generic Elective Courses		—
	Semester-X (Total Credits)		**24**

TEACHING & EXAMINATION SCHEME FOR 5 YEAR INTEGRATED M. Sc. (COMPUTER SCIENCE) COURSE

Semester-X	Course		Hours per week								Credits
	No.	Name	Theory	Practical	Internal	Total	Lectures	Others (Tutorials)	Practical	Total	
Semester-X											
	MCSCC521	Major Project	***	400	200	600	***	2	44	46	24
		Total	***	400	200	600	***	2	44	46	24

Theory/Tutorial Credits → 1 Hour = 1 Credit

Practical Credits → 2 Hours = 1 Credit

Course Name: Major Project

Course Code: MCSCC521

Objectives:

The aim of this course is

- To introduce structured development of software systems
- To acquaint students to various techniques of requirements determination
- To introduce the concepts of analysis and design
- To model the system with various software diagrams
- To develop a complete system/module of a large system using software engineering concepts
- To prepare document/report of the system

Prerequisites:

Knowledge of Programming

Contents:

1. **Introduction to System Analysis & Design**
   Concepts of System Analysis & Design, Components of a System, Users of a System, System Development Life Cycle, Role of System Analyst, Various approaches to develop a system, Using a “systems” approach to build whole system

2. **Structured Systems Development Approach**
   Requirements Gathering, surveying existing systems, Performing Feasibility Study before building a system, Formulating SRS, Designing Data Flow Diagrams, Entity-Relationship Diagrams, Designing Data Dictionary, Designing effective Input and Reports, Implementing EOD, EOM and EOY Procedures, Documenting the system

Guidelines for Project:

1. Projects can be created by a single student or in a group of 2-3 students
2. The definition of project is to be submitted within 15 days of the starting of the semester
3. The project should be free from plagiarism
4. A student/group is required to report to their project guides on regular basis
5. A student/group is required to present the project during internal exams
6. It is advisable that the project must include different forms and reports
7. It is necessary to prepare system diagrams, data dictionary as per the convention
8. Project documentation should be prepared for the semester end examination
9. In the semester end examination, the project should be presented to the examiners along with the documentation(report)

References:

1. Pressman R., Maxim B., “Software Engineering – A practioner’s approach”, McGraw Hill
2. Kendall & Kendall. “System Analysis and Design ”, Prentice Hall India (PHI)
3. James Sen, “ Analysis and Design of Information System”, PHI
4. Arthur Langer, “ Analysis and Design of Information Systems “, Springer
5. Jeff Johnson, “ Designing with the Mind in Mind “, Morgan Kaufmann
6. Dennis, Wixom, Roth “System Analysis and Design”, Wiley India
7. Joseph Valacich, Joey F. George, “Modern Systems Analysis and Design”, Pearson
8. Priti Srinivas Sajja., “Essence of System Analysis and Design”, Springer

Accomplishments of the student after completing the Course:

After completion of this course, students will be able to

- Analyse and Design a system based on user requirements
- develop a software system using software engineering concepts
- prepare a document for the system
