MANGALORE UNIVERSITY

Scheme of Examination and Syllabus for

Master of Science in Computer Science Degree Programme

Choice Based Credit System (CBCS)

DEPARTMENT OF POST-GRADUATE STUDIES AND RESEARCH IN COMPUTER SCIENCE
MANGALAGANGOTHRI-574 199

PROGRAMME LEARNING OBJECTIVES (LOs)

M.Sc. Computer Science Degree programme provides a rigorous education that combines central topics in computing and specialization in a more focused area with added emphasis on the physical and architectural underpinnings of modern computer system design. Our graduates have the breadth of understanding a practice both in traditional areas of computing and in applications to other disciplines. The Learning objectives of this programme are:

PLO1: Practice and grow as computing professionals, conducting research and/or leading, designing, developing or maintaining projects in various technical areas of computer science.
PLO2: Utilize knowledge and skills in Computer Science effectively for improving the society.
PLO3: Use new technical advancements of Computer Science to produce tangible contributions in the profession

PROGRAMME OUTCOMES (LOs)

The curriculum leading to M.Sc-Computer Science degree prepares the students for the positions as computer scientists, Data scientists, and software engineers and Academicians in Business Intelligence, Information Technology, Software Industry and Government sectors. The curriculum's main objectives are to impart students with an understanding of the Hardware, Software and problem solving skills through Algorithmic approaches and to develop proficiency in the practice of computing, and to prepare them for continued professional development.

Upon completion of M.Sc.-Computer Science, students will be able to:

PO1: Apply algorithmic, mathematical and scientific reasoning to a variety of computational problems
PO2: Design, Analyze, implement and document solutions to significant computational problems
PO3: Analyze and compare alternative solutions to computing problems
PO4: Implement software systems that meet specified design and performance requirements
PO5: Work effectively in teams to design and implement solutions to computational problems
PO6: Communicate effectively, both orally and in writing
PO7: Recognize the social and ethical responsibilities of a professional working in the discipline

PROGRAMME SPECIFIC OUTCOMES (PSO)

On completion of the M.Sc.-Computer Science Degree programme the graduates will be able to

PSO1: Design and develop computer programs/computer-based systems in the areas related to algorithms, networking, web design, cloud computing, IoT and data analytics of varying complexity.
PSO2: Apply standard Software Engineering practices and strategies in real-time software project development using open-source programming environment or commercial environment to deliver quality product for the organization success.
PSO3: Acquaint with the contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems.

I SEMESTER M.Sc. Computer Science								
Subject Code	Subjects	Theory Hours/ Week	Practical Hours/ Week	Duration of exams (Hrs)	Marks & Credits			
					IA	Exam	Total	Credits
HARD CORE								
CSH401	**Foundations of Computer Science**	4L	-	3	30	70	100	4
CSH402	**Algorithmics**	4L	-	3	30	70	100	4
CSH403	**Data Communications and Computer Networks**	4L	-	3	30	70	100	4
CSH404	**Embedded Systems**	4L	-	3	30	70	100	4
SOFT CORE								
CSS405	**JAVA Technology**	4L	-	3	30	70	100	4
CSP406	**Algorithmics Lab**	-	6	3	30	70	100	3
CSP407	**Java Programming Lab**	-	6	3	30	70	100	3
	Total	20	12	21	210	490	700	26

II SEMESTER M.Sc. Computer Science								
Subject Code	Subjects	Theory Hours/ Week	Practical Hours/ Week	Duration of exams(Hrs)	Marks & Credits			
					IA	Exam	Total	Credits
HARD CORE								
CSH451	**Advanced Operating System**	4L	-	3	30	70	100	4
CSH452	**Internet of Things**	4L	-	3	30	70	100	4
CSH453	**Advanced Database Management Systems**	4L	-	3	30	70	100	4
SOFT CORE								
CSS454 **CSS455** **CSS456** **CSS457** **CSS458**	**Data Science** **Mobile & Wireless Communications** **Machine Learning** **Natural Language Processing** **Object-Oriented Data Modeling Using UML**	4L	-	3	30	70	100	4
CSP459 **CSP460** **CSP461**	**Operating System Lab** **Data Science Lab** **Machine Learning Lab**	---	6	3	30	70	100	3
CSP462 **CSP463** **CSP464**	**DBMS Lab** **Natural Language Processing Lab** **Internet of Things Lab**	---	6	3	30	70	100	3
OPEN ELECTIVE								
CSE465	**Web Technologies**	3L	--	3	30	70	100	3*
CSE466	**Linux Environment Systems**							
	Total Semester	**20**	**12**	**21**	**210**	**490**	**700**	**22+3***

*** Not included for CGPA.**

III SEMESTER M.Sc. Computer Science								
Subject Code	Course	Theory Hours / Week	Practical Hours / Week	Duration of exams (Hrs)	Marks & Credits			
					IA	Exam	Total	Credits
HARD CORE								
CSH501	**.NET Technology**	4L	-	3	30	70	100	4
CSH502	**Computer Graphics and Multimedia**	4L	-	3	30	70	100	4
CSH503	**Software Engineering**	4L	-	3	30	70	100	4
SOFT CORE								
CSS504 **CSS505** **CSS506** **CSS507** **CSS508**	**Information Retrieval** **Big Data Analytics** **Android Applications** **Digital Image Processing** **Cloud Computing**	4L	-	3	30	70	100	4
CSP509 **CSP510** **CSP511**	**NET Programming Lab** **Big Data Analytics Lab** **Android Applications Lab**	---	6	3	30	70	100	3
CSP512 **CSP513** **CSP514**	**Computer Graphics Lab** **Image Processing Lab** **Cloud Computing Lab**	---	6	3	30	70	100	3
OPEN ELECTIVE								
CSE515	**Open Source Technologies**	3L		3	30	70	100	3*
CSE516	**Mobile E-Commerce**							
	Total	**19**	**12**	**21**	**210**	**490**	**700**	**22+3***

*** Not included for CGPA.**

IV SEMESTER M.Sc. Computer Science							
Subject Code	Subject	Practical Hours/ Week	Duration of exams (Hrs)	Marks & Credits			
				IA	Dissertation + Viva Exam	Total	Credits
CSP 551	Project Work Report Viva-Voce	32	-	120	280 (Report : 180 Viva-Voce: 100)	400	16
Total Marks off I Semester						700	26
Total Marks off II Semester						700	22+3*
Total Marks off III Semester						700	22+3*
Total Marks of IV Semester						400	16
Grand Total Marks & Credits of all the Four Semesters						2500	86+6*

Project Work at Software Company / University/ National Institute

Hard core Credits: 16 + 12 + 12 + 16 = 56 (60.87%)

Total Soft-Core Credits: 10 +10+10 = 30 (32.60%)

Open Elective Credits: 03* +03* = 06 (6.52%)

***Not included for CGPA.**

CSH 401: FOUNDATIONS OF COMPUTER SCIENCE

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Basic probability axioms and rules and the moments of discrete and continuous random variables as well as be familiar with common named discrete and continuous random variables.
CO2: Will apply knowledge of computing and mathematics appropriate to the discipline.
CO3: Will function effectively as a member of a team in order to accomplish a common goal.
CO4: Will apply mathematical foundations, algorithmic principles and computer science theory to the modeling and design of computer based systems in a way that demonstrates
CO5: Will apply design and development principles in the construction of software systems of varying complexity

UNIT-I **12 Hrs.**

Review of Sets, Propositions, Relations, Functions, Graphs, *Introduction to Probability theory:* Introduction, Sample space-random variables - probability distributions, expected values, joint distributions, variance, covariance

UNIT-II **12 Hrs.**

Secondary Storage: Introduction, classification, magnetic tape, magnetic disk, Optical disk, ***Theory of Computation:*** *Introduction:* Strings and their properties, Formal Languages, Types of Grammars and Languages, Chomsky classification of Languages, Recursive and recursively enumerable sets, Operations.
Theory of Automata: Finite State Models, Minimization, Regular sets and Regular Grammars, Pumping Lemma, Closure properties, Applications of Finite automata.

UNIT-III **12 Hrs.**

***Context Free Languages*:** Context Free Grammar and Push Down Automata, equivalence of PDA and CFG, Deterministic PDA, Normal forms, Applications of CFG

UNIT-IV **12 Hrs.**

Turing machines and Linear Bounded Automata: TM model, Representation and Design of TM, Halting problem, Universal TM and modifications, Linear bounded automata.

REFERENCE BOOKS:

1. JD Ullman et al. ,Introduction to Automata Theory, Languages and Computation, 3rd Edition, Pearson Publication, 2006.
2. C L Liu, Elements of Discrete Mathematics: A Computer Oriented Approach, McGraw-Hill edition, 2013.
3. P K. S. Trivedi robability and Statistics with Reliability, Queuing and Computer Science Applications, First Edition, Prentice Hall of India.,,2008
4. **Schöning**, Uwe, **Pruim**, Randall J, Gems of Theoretical Computer Science, Springer Publications.
5. Hary R Lewis, Christor H Papadi metrion ,Elements of the Theory of Computation, Prentice-Hall International, 1998.
6. KLP Mishra and N Chandrashekar,,Theory of Computer Science, 3rd Edition, PHI publication,2007.

CSH 402: ALGORITHMICS

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Argue the correctness of algorithms using inductive proofs and invariants.

CO2: Analyze worst-case running times of algorithms using asymptotic analysis.

CO3: Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.

CO4: Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms, and analyze them.

CO5: Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize greedy algorithms, and analyze them.

CO6: Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.

CO7: Explain the different ways to analyze randomized algorithms (expected running time, probability of error). Recite algorithms that employ randomization. Explain the difference between a randomized algorithm and an algorithm with probabilistic inputs.

UNIT-I **12 Hrs**

Introduction: Algorithms, performance analysis-time complexity and space complexity, O-notation, Omega notation and Theta notation, Review of basic data structures , priority queues- , heaps, definition, insertion and deletion, application-heap sort, Introduction to Skip List, skip list representation, operations- insertion, deletion and searching , Hashing, hash table representation, hash functions, collision resolution-separate chaining, open addressing-linear probing, quadratic probing, double hashing and comparison of hashing and skip lists.

UNIT-II **12 Hrs.**

Search Trees: Binary Search Trees, definition, ADT, implementation, operations-searching, insertion and deletion, Balanced search trees- AVL trees, definition, height of an AVL tree, representation, operations-insertion, deletion and searching. Introduction to Red – Black trees and Splay Trees, B-Trees, insertion, deletion and searching, Comparison of Search Trees.

UNIT-III **12 Hrs.**

Divide and Conquer: General Method – Binary Search – Finding Maximum and Minimum – Merge Sort , **Greedy method**: General method, Minimum cost spanning trees, Job sequencing with deadlines, **Backtracking:** General Method – 8 Queens problem – sum of subsets – graph coloring – Hamiltonian problem – knapsack problem.

UNIT-IV **12 Hrs.**

Dynamic Programming: General method, Optimal binary search trees, 0/1 knapsack problem, Travelling sales person problem. **Graphs:** Graph Traversals – Connected Components – Spanning Trees – Biconnected components – Branch and Bound: General Methods (FIFO & LC) – 0/1 Knapsack problem – Introduction to NP-Hard and NP-Completeness.

REFERENCE BOOKS:

1. Mark A. Weiss, “Data structures and Algorithm analysis in C++(Java)”, Fourth Edition, PHI ,2013
2. Michael T. Goodrich, R. Tamassia and D. Mount “Data structures and Algorithms in C++”, Wiley student edition, John Wiley and Sons.
3. Data Structures and Algorithms in C++, Second Edition, Adam Drozdek, Vikas Publishing House, Thomson International Student Edition.
4. **[4]** Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, “Computer Algorithms/ C++”, Second Edition, Universities Press, 2007.
5. Horowitz and Sahni, and Rajashekaran, Fundamentals of Computer Algorithms, University Press, 2nd Edition, Galgotia Publications, 2007
6. Anany Levitin ,Introduction to the Design & Analysis of Algorithms, Pearson Addison-Wesley, 2007 .

CSH 403:DATA COMMUNICATIONS AND COMPUTER NETWORKS

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Explain how communication works in data networks and the Internet.
CO2: Recognize the different internetworking devices and their functions.
CO3: Explain the role of protocols in networking.
CO4: Analyze the services and features of the various layers of data networks.
CO5: Design, calculate, and apply subnet masks and addresses to fulfill networking requirements.

UNIT-I **12 Hrs.**

Introduction: Data communications fundamentals, computer communications architecture, Data Communication tasks, Data Communication Systems Applications, Data Communication System Characteristics features, Data Communication network criteria, Protocols and standards, Transmission mode, Analog and Digital Signals, Bit rate, Baud rate, Channel capacity using Nyquist and Shannon's relation. Modulation, encoding and decoding techniques. Transmission media characteristics, Transmission impairments, multiplexing.

UNIT-II **12 Hrs.**

Introduction to Computer Networks, Application and goals, Classification of Computer Networks, ISO-OSI Architecture, Services of Physical, Data link, Network, Transport, Session, Presentation and Application Layers., TCP /IP reference Model, Topology. Physical and Data Link Layer Services, Network Layer Services: Networking and Internetworking Technology Devices, Repeaters, Bridges, Routers, Gateways and Other Devices.

UNIT-III **12 Hrs.**

TCP/IP Protocol Suit: Overview of TCP/IP, TCP/IP and the Internet, TCP/IP and OSI, Internetwork Protocol (IP), Classes of IP, Addressing, Protocols in the Network Layer, Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), Internet Control MESSAGE Protocol (ICMP), Internet Group Message Protocol (IGMP), Transport Layer Services, Functionalities of the Transport Layer.

UNIT-IV **12 Hrs.**

Upper OSI Layers: Session Layer Services, SPDU. Presentation Layer Services: Application layer Services, PPDU. Application Layer Services: Client / Server Model,, BOOTP, Dynamic Host Configuration Protocol(DHCP), Domain Name System (DNS), Telnet, Fle transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP),Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Simple Network Management Protocol (SNMP), Hyper Text Transfer Protocol (HTTP) , World Wide Web (WWW).

REFERENCE BOOKS:

1. Prakash C. Gupta, Data Communications and Computer Networks, PHI (Latest Edition), 2013.
2. Behrouz A Forouzan, Data Communications and Networking, McGraw Hill, (Fourth Edition), 2007.
3. Behrouz A Forouzan and Firouz, Computer Networks A Top - Down Approach, McGraw Hill, (Special Indian Edition), 2012.
4. Tananbaum A.S., "Computer Networks", 3rd Ed, PHI, 1999.
5. Black U., "Computer Networks-Protocols, Standards and Interfaces", PHI, 1996.
6. Stallings W., "Computer Communication Networks", PHI.
7. Stallings W., "SNMP, SNMPv2, SNMPv3, RMON 1&2", 3rd Ed., Addison Wesley, 1999.

8. Michael A. Miller, “Data & Network Communications”, Vikas Publication, 2008
9. William A. Shay, “Understanding Data Communications & Networks”, Vikas Publication,2008.

CSH 404: EMBEDED SYSTEMS

Hours/Week: 4 **I.A. Marks: 30**

Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Understand what is a microcontroller, microcomputer, embedded system.
CO2: Understand different components of a micro-controller and their interactions.
CO3: Become familiar with programming environment used to develop embedded systems.
CO4: Understand key concepts of embedded systems like IO, timers, interrupts, interaction with peripheral devices.
CO5: Learn debugging techniques for an embedded system.

UNIT-I **12 Hrs.**

INTRODUCTION TO EMBEDDED SYSTEMS: Embedded systems; Processor embedded into a system; Embedded hardware units and devices in a system; Embedded software in a system; Examples of embedded systems; Embedded System-on-Chip (SoC) and use of VLSI circuit design technology; Complex systems design and processors; Design process in embedded system; Formalization of system design; Design process and design examples; Classification of embedded systems; I/O types and examples; Serial communication devices; Parallel device ports

UNIT-II **12 Hrs.**

COMMUNICATION BUSES FOR DEVICE NETWORKS: Wireless devices; Timer and counting devices; Watchdog timer; Real time clock; Networked embedded systems; Serial bus communication protocols; Parallel bus device protocols; Internet enabled systems; Wireless and mobile system protocols; Device access without interrupts; ISR concept; Interrupt sources; Interrupt servicing mechanism; Multiple interrupts; Context and the periods for context-switching, interrupt latency and deadline; Classification of processors' interrupt service mechanism from context-saving angle; Direct Memory Access; Device drivers programming.

UNIT-III **12 Hrs.**

PROGRAM MODELING CONCEPTS, PROCESSES, THREADS, AND TASKS: Program models; DFG models; State machine programming models for event controlled program flow; Modeling of multiprocessor systems. Multiple processes in an application; Multiple threads in an application; Tasks and task states; Task and data; Distinctions between functions, ISRs and tasks.

OPERATING SYSTEMS: Operating System services; Process management; Timer functions; Event functions; Memory management; Device, file and I/O sub-systems management; Interrupt routines in RTOS environment and handling of interrupt source calls.

UNIT-IV **12 Hrs.**

REAL-TIME OPERATING SYSTEMS: Real-Time Operating Systems; Basic design using an RTOS; RTOS task scheduling models, interrupt latency and response times of the tasks as performance metrics; OS security issues.

EMBEDDED SOFTWARE DEVELOPMENT, TOOLS: Introduction; Host and target machines; Linking and locating software; Getting embedded software in to the target system; Issues in hardware-software design and co-design; Testing on host machine; Simulators; Laboratory tools.

REFERENCE BOOKS:

1. Rajkamal, **Embedded Systems Architecture: Programming and Design –** Tata McGraw Hill, 2nd Edition, 2008.
2. Wayne Wolf,**, Computers as Components: Principles of Embedded Computer System Design** –Elsevier, 2005.
3. Tammy Noergaard, **Embedded Systems Architecture** –Elsevier, 2005.

4. Steve Heath,**, Embedded Systems Design,** 2nd Edition, Elsevier, 2003.
5. Dr. K.V.K.K. Prasad,, **Embedded/Real-Time Systems: Concepts, Design and Programming: The Ultimate Reference,** Dreamtech Press, 2004.
6. Michael J.Point, **Embedded C,** Pearson Education, 2002.

CSS 405: JAVA TECHNOLOGY

Hours/Week: 4 **I.A. Marks: 30**

Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Be able to understand the difference between object oriented programming and procedural oriented language.

CO2: Be able to understand and implement data types in Java

CO3: Be able to program using Java features such as composition of objects, Operator overloading, inheritance, Polymorphism etc.

CO4: At the end of the course students will able to simulate the problem in the subjects like Operating system, Computer networks.

CO5: Students will able to simulate real world problems.

UNIT-I **12 Hrs.**

INTRODUCTION TO JAVA: Java and Java applications; Java Development Kit (JDK); Java is interpreter, Byte Code, JVM; Object-oriented programming; Simple Java programs. Data types and other tokens; Creating and destroying objects; Access specifies; Operators and Expressions; Control Statements: Selection statements, iteration statements, Jump Statements.

UNIT-II **12 Hrs.**

CLASSES, INHERITANCE, EXCEPTIONS, APPLETS: Classes: Classes in Java; Declaring a class; Class name; Super classes; Constructors; Creating instances of class; Inner classes. Inheritance: Simple, multiple, and multilevel inheritance; Overriding, overloading. Exception handling**:** Exception handling in Java. The Applet Class: Two types of Applets; Applet basics; Applet Architecture; An Applet skeleton; Simple Applet display methods; Requesting repainting; Using the Status Window; The HTML APPLET tag; Passing parameters to Applets; getDocumentbase() and getCodebase(); ApletContext and showDocument(); The AudioClip Interface; The AppletStub Interface; Output to the Console

UNIT-III **12 Hrs.**

MULTI THREADED PROGRAMMING, EVENT HANDLING: Multi Threaded Programming: What are threads? How to make the classes threadable; Extending threads; Implementing runnable; Synchronization; Changing state of the thread; Bounded buffer problems, read-write problem, producer-consumer problems. Event Handling: Two event handling mechanisms; The delegation event model; Event classes; Sources of events; Event listener interfaces; Using the delegation event model; Adapter classes; Inner classes.

UNIT-IV **12 Hrs.**

SWINGS: Swings: The origins of Swing; Two key Swing features; Components and Containers; The Swing Packages; A simple Swing Application; Create a Swing Applet; Jlabel and ImageIcon; JTextField; The Swing Buttons; JTabbedpane; JScrollPane; JList; JComboBox; JTable.

JAVA 2 ENTERPRISE EDITION OVERVIEW, DATABASE ACCESS: Overview of J2EE and J2SE. The Concept of JDBC; JDBC Driver Types; JDBC Packages; A Brief Overview of the JDBC process; Database Connection; Associating the JDBC/ODBC Bridge with the Database; Statement Objects; ResultSet; Transaction Processing; Metadata, Data types; Exceptions.

REFERENCE BOOKS:

1. Herbert Schildt, **Java - The Complete Reference,** 7th Edition, Tata Mcgraw Hill, 2007. Jim Keogh, **J2EE - The Complete Reference,** Tata Mcgraw Hill, 2007.
2. Y. Daniel Liang, **Introduction to JAVA Programming,** 6th Edition, Pearson Education, 2007.
3. Stephanie Bodoff et al, **The J2EE Tutorial,** 2nd Edition, Pearson Education, 2004

CSP 406: Algorithmic Lab

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1: Understand the object oriented concepts for implementation.
CO2: Implement the data structure concepts.
CO3: Employ good software engineering practices such as incremental development, data integrity checking and adherence to style guidelines.
CO4: Select and model data using primitive and structured types.
CO5: Construct programs that demonstrate effective use of C features including arrays, structures, pointers and files.

CSP 407: Java Programming Lab

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1: Understand the Java programming language
CO2: Understand the aspects of designing.
CO3: Understand coding and implementation.
CO4: Know about new ideas and advances.
CO5: Know about techniques, and tools and to use them effectively.
CO6: Ability to design and implement websites and desktop applications.

*** Not included for CGPA.**

II SEMESTER

CSH 451: ADVANCED OPERATING SYSTEM

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Analyze the structure of OS and basic architectural components involved in OS design
CO2: Analyze and design the applications to run in parallel either using process or thread models of different OS
CO3: Analyze the various device and resource management techniques for timesharing and distributed systems
CO4: Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system
CO5: Interpret the mechanisms adopted for file sharing in distributed Applications
CO6: Conceptualize the components involved in designing a contemporary OS

UNIT-I **12 Hrs.**

Operating System Overview : Operating System Objectives and Functions, The Evolution of Operating Systems, Major Achievements, Developments Leading to Modern Operating Systems, Microsoft Windows Overview, Traditional UNIX Systems, Modern UNIX Systems, Linux. **Process description & control :** What is a Process?, Process States, Process Description, Process Control, Execution of the Operating System, Security Issues, UNIX SVR4 Process Management.

UNIT-II **12 Hrs.**

Threads, SMP, and Microkernel: Processes and Threads, Symmetric Multiprocessing (SMP), Microkernels, Windows Vista Thread and SMP Management, Solaris Thread and SMP Management, Linux Process and Thread Management..
Virtual Memory : Hardware and Control Structures, Operating System Software, UNIX and Solaris Memory Management, Linux Memory Management, Windows Vista Memory Management, Summary.

UNIT-III **12 Hrs.**

Multiprocessor and Real-Time Scheduling: Multiprocessor Scheduling, Real-Time Scheduling, Linux Scheduling, UNIX PreclsSl) Scheduling, Windows Vista Scheduling.
Distributed Process Management: Process Migration, Distributed Global States, Distributed Mutual Exclusion, Distributed Deadlock. **Security:** Security Threats, Attacks, and Assets, Intruders, Malicious Software Overview, Viruses, Worms, and Bots, Rootkits

UNIT-IV **12 Hrs.**

Kernel Organization: Using Kernel Services, Daemons, Starting the Kernel, Control in the Machine, Modules and Device Management, Module Organization, Module Installation and Removal, Process and Resource Management, Running Process Manager, Creating a new Task, IPC and Synchronization, The Scheduler, Memory Manager, The Virtual Address Space, The Page Fault Handler, File Management.
The windows NT/2000/XP kernel: Introduction, The NT kernel, Objects, Threads, Multiplication Synchronization, Traps, Interrupts and Exceptions, The NT executive , Object Manager, Process and Thread Manager, Virtual Memory Manager, I/o Manager, The cache Manager , Kernel local procedure calls and IPC, The native API, subsystems.

REFERENCE BOOKS:

1. William Stallings: Operating Systems: Internals and Design Principles, 6th Edition, Prentice Hall, 2013.
2. Gary Nutt: Operating Systems, 3rd Edition, Pearson, 2014.
3. Silberschatz, Galvin, Gagne: Operating System Concepts, 8th Edition, Wiley, 2008
4. Andrew S. Tanenbaum, Albert S. Woodhull: Operating Systems, Design and Implementation, 3rd Edition, Prentice Hall, 2006.
5. Pradeep K Sinha: Distributed Operating Systems, Concept and Design, PHI, 2007.

CSH 452: INTERNET OF THINGS

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Apply the concepts of IOT.
CO2: Apply IOT to different applications.
CO3: Analysis and evaluate protocols used in IOT.
CO4: Design and develop smart city in IOT.
CO5: Analysis and evaluate the data received through sensors in IOT.

UNIT-I **12 Hrs.**

Introduction to Internet of Things –Definition and Characteristics of IoT, Physical Design of IoT – IoT Protocols, IoT communication models, Iot Communication APIs IoT enabaled Technologies – Wireless Sensor Networks, Cloud Computing, Big data analytics, Communication protocols, Embedded Systems, IoT Levels and Templates Domain Specific IoTs – Home, City, Environment, Energy, Retail, Logistics, Agriculture, Industry, health and Lifestyle

UNIT-II **12 Hrs.**

IoT and M2M – Software defined networks, network function virtualization, difference between SDN and NFV for IoT Basics of IoT System Management with NETCOZF, YANG-NETCONF, YANG, SNMP NETOPEER

UNIT-III **12 Hrs.**

Introduction to Python - Language features of Python, Data types, data structures, Control of flow, functions, modules, packaging, file handling, data/time operations, classes, Exception handling Python packages - JSON, XML, HTTPLib, URLLib, SMTPLib. IoT Physical Devices and Endpoints - Introduction to Raspberry PI-Interfaces (serial, SPI, I2C) Programming – Python program with Raspberry PI with focus of interfacing external gadgets, controlling output, reading input from pins.

UNIT-IV **12 Hrs.**

IoT Physical Servers and Cloud Offerings – Introduction to Cloud Storage models and communication APIs Webserver – Web server for IoT, Cloud for IoT, Python web application framework Designing a RESTful web API.

REFERENCE BOOKS:

1. Arshdeep Bahga and Vijay Madisetti,,Internet of Things - A Hands-on Approach, Universities Press, 2015, ISBN: 9788173719547
2. Matt Richardson & Shawn Wallace, Getting Started with Raspberry Pi, O'Reilly (SPD), 2014, ISBN: 9789350239759.

CSH 453: ADVANCEDD DATABASE SYSTEMS

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Explain the features of database management systems and Relational database.
CO2: Design conceptual models of a database using ER modelingfor real life applications and also construct queries in Relational Algebra.
CO3: Create and populate a RDBMS for a real life application, with constraints and keys, using SQL.
CO4: Retrieve any type of information from a data base by formulating complex queries in SQL.
CO5: Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database.
CO6: Build indexing mechanisms for efficient retrieval of information from a database.

UNIT-I **12 Hrs.**

OBJECT AND OBJECT RELATIONAL DATABASES: Concepts for Object Databases: Object Identity – Object structure – Type Constructors –Encapsulation of Operations – Methods – Persistence – Type and Class Hierarchies – Inheritance – Complex Objects – Object Database Standards, Languages and Design: ODMG Model – ODL –OQL – Object Relational and Extended – Relational Systems: Object Relational features in SQL/Oracle – Case Studies.

UNIT-II **12 Hrs.**

PARALLEL AND DISTRIBUTED DATABASES : Database System Architectures: Centralized and Client-Server Architectures – Server SystemArchitectures – Parallel Systems-Distributed Systems – Parallel Databases: I/O Parallelism – Interand Intra Query Parallelism – Inter and Intra operation Parallelism – Design of Parallel Systems-Distributed Database Concepts - Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing – Case Studies

UNIT-III **12 Hrs.**

INTELLIGENT DATABASES: Active Databases: Syntax and Semantics (Starburst, Oracle, DB2)- Taxonomy- Applications Design Principles for Active Rules- Temporal Databases: Overview of Temporal Databases-TSQL2- Deductive Databases: Logic of Query Languages – Datalog- Recursive Rules-Syntax and Semantics of Datalog Languages- Implementation of Rules and Recursion- Recursive Queries in SQL- Spatial Databases- Spatial Data Types-Spatial Relationships- Spatial Data Structures Spatial Access Methods-Spatial DB Implementation.

UNIT-IV **12 Hrs.**

ADVANCED DATA MODELS : Mobile Databases: Location and Handoff Management - Effect of Mobility on Data Management -Location Dependent Data Distribution - Mobile Transaction Models -Concurrency Control -Transaction Commit Protocols- Multimedia Databases- Information Retrieval- Data Warehousing-Data Mining- Text Mining.

EMERGING TECHNOLOGIES : XML Databases: XML-Related Technologies-XML Schema- XML Query Languages- Storing XML in Databases-XML and SQL- Native XML Databases- Web Databases- Geographic Information Systems- Biological Data Management-Cloud Based Databases: Data Storage Systems on the Cloud- Cloud Storage Architectures-Cloud Data Models- Query Languages- Introduction to Big Data-Storage-Analysis.

REFERENCE BOOKS:

1. Elmasri and Navathe, **Fundamentals of Database Systems** 5th Edition, Addison-Wesley, 2007.
2. Raghu Ramakrishnan and Johannes Gehrke**, Database Management Systems** ,3rd Edition, McGraw-Hill, 2003.
3. **Data Base System Concepts,** Silberschatz, Korth and Sudharshan, 5th Edition, Mc-Graw Hill, 2006.
4. C.J. Date, A. Kannan, S. Swamynatham**, An Introduction to Database Systems**, 8th Edition, Pearson Education, 2006.

SOFT ELECTIVE-I
IN
II SEMESTER

CSS 454: DATA SCIENCE

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Students will develop relevant programming abilities.
CO2: Students will demonstrate proficiency with statistical analysis of data.
CO3: Students will develop the ability to build and assess data-based models.
CO4: Students will execute statistical analyses with professional statistical software.
CO5: Students will demonstrate skill in data management.

UNIT-I **12 Hrs.**

Introduction: The Ascendance of Data, What Is Data Science?, Motivating Hypothetical:, Finding Key Connectors, Data Scientists You May Know, Salaries and Experience, Paid Accounts, Topics of Interest, Onward. **Python:** The Basics: Getting Python, The Zen of Python, Whitespace Formatting, Modules, Arithmetic, Functions, Strings, Exceptions, Lists, Tuples, Dictionaries, Sets, Control Flow, Truthiness; The Not-So-Basics: Sorting, List Comprehensions, Generators and Iterators, Randomness, Regular Expressions, Object-Oriented Programming, Functional Tools, Enumerate, zip and Argument Unpacking, args and kwargs. **Visualizing Data:** matplotlib, Bar Charts, Line Charts, Scatterplots. **Linear Algebra:** Vectors, Matrices.

UNIT-II **12 Hrs.**

Statistics: Describing a Single Set of Data: Central Tendencies, Dispersion. Correlation, Simpson's Paradox, Some Other Correlation Caveats, Correlation and Causation. **Probability:** Dependence and Independence, Conditional Probability, Bayes's Theorem, Random Variables, Continuous Distributions, The Normal Distributions, The Central Limit Theorem. **Hypothesis and Inference:** Statistical Hypothesis Testing, Example: Flipping a Coin, Confidence Intervals, P-hacking, Example: Running an A/B Test, Bayesian Inference. **Gradient Descent:** The Idea Behind Gradient Descent, Estimating the Gradient, Using the Gradient, Choosing the Right Step Size, Putting It All Together, Stochastic Gradient descent. **Getting Data:** stdin and stdout, Reading Files: The Basics of text Files, Delimited Files. Scraping the Web: HTML and the Parsing Thereof, Example: O'Reilly Books About Data. Using APIs: JSON(and XML) Using an Unauthenticated API, Finding APIs Example: Using the Twitter APIs, Getting Credentials. **Working with Data:** Exploring Your Data: Exploring One-Dimensional Data, Two Dimensions, Many Dimensions; Cleaning and Munging, Manipulating Data, Rescaling, Dimensionality Reduction.

UNIT-III **12 Hrs.**

Machine Learning: Modeling, What Is Machine Learning? Overfitting and Underfitting, Correctness, The Bias-Variance Trade-off, Feature Extraction and Selection. **k-Nearest Neighbors:** The Model, Example: Favorite Languages, The Curse of Dimensionality. **Naive Bayes:** A Really Dumb Spam Filter, A More Sophisticated Spam Filter, Implementation, Testing Our Model. **Simple Linear Regression:** The Model, Using Gradient Descent, Maximum Likelihood Estimation. **Multiple Regression:** The Model, Further Assumptions of the Least Squares Model, Fitting the Model, Goodness of Fit, Digression: The Bootstrap, Standard Errors f Regression Coefficients, Regularization. **Logistic Regression:** The Problem, The Logistic Function, Applying the Model, Goodness of Fit, Support Vector Machines. **Decision Trees:** What Is a Decision Tree? Entropy, The Entropy of a Partition, Creating a Decision Tree, Putting It All Together, Random Forests.

UNIT-IV **12 Hrs.**

Neural Networks: Perceptrons, Feed-Forward Neural Networks, Backpropagation, Example: Defeating a CAPTCHA. **Clustering:** The Idea, The Model, Example: Meetups, Choosing k,

Example: Clustering Colors, Bottom-up Hierarchical Clustering. **Natural Language Processing:** Word Clouds, n-gram Models, Grammars, An Aside: Gibbs Sampling, Topic Modeling. **Network Analysis:** Betweenness Centrality, Eigenvector Centrality: Matrix Multiplication, Centrality; Directed Graphs and PageRank. **Recommender Systems:** Manual Curation, Recommending What's Popular, User-Based Collaborative Filtering, Item-Based Collaborative Filtering. **Database and SQL:** CREATE TABLE and INSERT, UPDATE, DELETE, SELECT, GROUP BY,ORDER BY, JOIN, Subqueries, Indexes, Query Optimization, NoSQL, **MapReduce:** Example: Word Count, Why MapReduce? MapReduce More Generally, Example: Analyzing Status Updates, Example: Matrix Multiplication, An Aside: Combiners.

REFERENCE BOOK:

1. Joel Grus, Data Science from Scratch: First Principles with Python, 1st Edition, ORIELLY Publications, 2015.
,
2. Rachel Schutt, Cathy O'NeilDoing Data Science: Straight Talk from the Frontline, 3rd Edition, O'Reilly Publication, 2014

CSS 455: MOBILE & WIRELESS COMMUNICATION

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Explain the basic concepts of wireless network and wireless generations.
CO2: Demonstrate the different wireless technologies such as CDMA, GSM, GPRS etc.
CO3: Appraise the importance of Ad-hoc networks such as MANET and VANET and Wireless Sensor networks
CO4: Describe and judge the emerging wireless technologies standards such as WLL, WLAN, WPAN, WMAN.
CO5: Explain the design considerations for deploying the wireless network infrastructure.
CO6: Differentiate and support the security measures, standards. Services and layer wise security considerations

UNIT-I **12 Hrs.**

Introduction to Personal Communications Services (PCS): PCS Architecture, Mobility management, Networks signaling. Global System for Mobile Communication (GSM) system overview: GSM Architecture, Mobility management, Network signaling.

UNIT-II **12 Hrs.**

General Packet Radio Services (GPRS): GPRS Architecture, GPRS Network Nodes. **Mobile Data Communication**: WLANs (Wireless LANs) IEEE 802.11 standard, Mobile IP.

UNIT-III **12 Hrs.**

Wireless Application Protocol (WAP): The Mobile Internet standard, WAP Gateway and Protocols, wireless mark up Languages (WML). **Third Generation (3G) Mobile Services**: Introduction to International Mobile Telecommunications 2000 (IMT 2000) vision, Wideband Code Division Multiple Access (W-CDMA), and CDMA 2000, Quality of services in 3G.

UNIT-IV **12 Hrs.**

Wireless Local Loop (WLL): Introduction to WLL Architecture, wireless Local Loop Technologies. **Global Mobile Satellite Systems**; case studies of the IRIDIUM and GLOBALSTAR systems. **Wireless Enterprise Networks**: Introduction to Virtual Networks, Blue tooth technology, Blue tooth Protocols. PAN, HAN, WPAN.

REFERENCE BOOKS

1. Yi-Bing Lin & Imrich Chlamtac,,Wireless and Mobile Networks Architectures, John Wiley & Sons, 2001.
2. Raj Pandya, Mobile and Personal Communication systems and services, Prentice Hall of India, 2001.
3. C Y Lee, Mobile Cellular Telecommunications; 2nd ed.; William, McGraw Hill
4. Kamilo Feher, Wireless and Digital Communications, Prentice-Hall, 1995.
5. Mark Ciampa,, Guide to Designing and Implementing wireless LANs, Thomson learning, Vikas Publishing House, 2001.
6. Ray Rischpater, Wireless Web Development, Springer Publishing, 2000.
7. Sandeep Singhal, "The Wireless Application Protocol", Pearson Education Asia, 2000.

CSS 456: MACHINE LEARNING

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Gain knowledge about basic concepts of Machine Learning
CO2: Identify machine learning techniques suitable for a given problem
CO3: Solve the problems using various machine learning techniques
CO4: Apply Dimensionality reduction techniques.
CO5: Design application using machine learning techniques.

UNIT-I **12 Hrs.**

INTRODUCTION - Well-posed learning problems, Designing a learning system, Perspectives and issues in machine learning Concept learning and the general to specific ordering – Introduction, A concept learning task, Concept learning as search, Find-S: finding a maximally specific hypothesis, Version spaces and the candidate elimination algorithm, Remarks on version spaces and candidate elimination, Inductive bias. Decision Tree learning – Introduction, Decision tree representation, Appropriate problems for decision tree learning, The basic decision tree learning algorithm, Hypothesis space search in decision tree learning, Inductive bias in decision tree learning, Issues in decision tree learning Artificial Neural Networks – Introduction, Neural network representation, Appropriate problems for neural network learning, Perceptions, Multilayer networks and the back propagation algorithm, Remarks on the back propagation algorithm, An illustrative example face recognition Advanced topics in artificial neural networks Evaluation Hypotheses – Motivation, Estimation hypothesis accuracy, Basics of sampling theory, A general approach for deriving confidence intervals, Difference in error of two hypotheses, Comparing learning algorithms.

UNIT-II **12 Hrs.**

Bayesian learning – Introduction, Bayes theorem, Bayes theorem and concept learning, Maximum likelihood and least squared error hypotheses, Maximum likelihood hypotheses for predicting probabilities, Minimum description length principle, Bayes optimal classifier, Gibs algorithm, Naïve Bayes classifier, An example learning to classify text, Bayesian belief networks The EM algorithm Computational learning theory – Introduction, Probability learning an approximately correct hypothesis, Sample complexity for Finite Hypothesis Space, Sample Complexity for infinite Hypothesis Spaces, The mistake bound model of learning - Instance-Based Learning- Introduction, k -Nearest Neighbour Learning, Locally Weighted Regression, Radial Basis Functions, Case-Based Reasoning, Remarks on Lazy and Eager Learning Genetic Algorithms – Motivation, Genetic Algorithms, An illustrative Example, Hypothesis Space Search, Genetic Programming, Models of Evolution and Learning, Parallelizing Genetic Algorithms.

UNIT-III **12 Hrs.**

Learning Sets of Rules – Introduction, Sequential Covering Algorithms, Learning Rule Sets: Summary, Learning First Order Rules, Learning Sets of First Order Rules: FOIL, Induction as Inverted Deduction, Inverting Resolution Analytical Learning - Introduction, Learning with Perfect Domain Theories: Prolog-EBG Remarks on Explanation-Based Learning, Explanation-Based Learning of Search Control Knowledge.

UNIT-IV **12 Hrs.**

Combining Inductive and Analytical Learning – Motivation, Inductive-Analytical Approaches to Learning, Using Prior Knowledge to Initialize the Hypothesis, Using Prior Knowledge to Alter the Search Objective, Using Prior Knowledge to Augment Search Operators, Reinforcement Learning – Introduction, The Learning Task, Q Learning, Non-Deterministic, Rewards and Actions, Temporal Difference Learning, Generalizing from Examples, Relationship to Dynamic Programming.

REFERENCE BOOKS:

1. Tom M. Mitchell, Machine Learning, McGraw-Hill Science,1997
2. Stephen Marsland, Machine Learning: An Algorithmic Perspective, Taylor & Francis (CRC), 2nd Edition
3. William W Hsieh, Machine Learning Methods in the Environmental Sciences, Neural Networks, Cambridge Univ Press, 2009
4. Richard O. Duda, Peter E. Hart and David G. Stork, pattern classification, John Wiley & Sons Inc.2nd edition, 2001
5. Chris Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
6. Peter Flach, Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2012

CSS 457: NATURAL LANGUAGE PROCESSING

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: After successful completion of this course, student will be able to

CO2: Understand approaches to syntax and semantics in NLP.

CO3: Understand approaches to discourse, generation, dialogue and summarization within NLP.

CO4: Understand current methods for statistical approaches to machine translation.

CO5: Understand machine learning techniques used in NLP, including hidden Markov models and probabilistic context-free grammars, clustering and unsupervised methods, log-linear and discriminative models, and the EM algorithm as applied within NLP

UNIT-I **12 Hrs.**

OVERVIEW AND LANGUAGE MODELING : *Overview*: Origins and challenges of NLP-Language and Grammar-Processing Indian Languages- NLP Applications-Information Retrieval. Language Modeling: Various Grammar- based Language Models-Statistical Language Model.

UNIT-II **12 Hrs.**

WORD LEVEL AND SYNTACTIC ANALYSIS : Word Level Analysis: Regular Expressions-Finite-State Automata-Morphological Parsing-Spelling Error Detection and correction-Words and Word classes-Part-of Speech Tagging. Syntactic Analysis: Context-free Grammar-Constituency-Parsing-Probabilistic Parsing.

UNIT-III **12 Hrs.**

SEMANTIC ANALYSIS AND DISCOURSE PROCESSING : Semantic Analysis: Meaning Representation-Lexical Semantics- Ambiguity-Word Sense Disambiguation. Discourse Processing: cohesion-Reference Resolution- Discourse Coherence and Structure.

UNIT-IV **12 Hrs.**

NATURAL LANGUAGE GENERATION AND MACHINE TRANSLATION :
Natural Language Generation: Architecture of NLG Systems- Generation Tasks and Representations- Application of NLG. Machine Translation: Problems in Machine Translation-Characteristics of Indian Languages- Machine Translation Approaches-Translation involving Indian Languages.

REFERENCE BOOKS:

1. Edward Loper, Ewan Klein, and Steven Bird, Natural Language Processing with Python, O'Reilly Publication 2009.;
2. Christopher D. Manning, Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT press,1999.
3. Dan Jurafsky, James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall, 2009.

CSS 458: OBJECT-ORIENTED ANALYSIS DESIGN WITH UML

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Demonstrate the ability to apply the knowledge of object oriented concepts for solving system modeling and design problems.
CO2: Design and implement object oriented models using UML appropriate notations.
CO3: Ability to apply the concepts of object oriented methodologies to design cleaner software from the problem statement.
CO4: Apply the concept of domain and application analysis for designing UML Diagrams.
CO5: Comprehend the concept of architectural design approaches for system design and implementation issues for object oriented models.
CO6: Illustrate the concept of patterns for constructing software architectures.

UNIT-I **12 Hrs.**

Introduction, Modeling Concepts, class Modeling

What is Object Orientation? What is OO development? OO themes; Evidence for usefulness of OO development; OO modeling history. **Modeling as Design Technique:** Modeling; abstraction; The three models. Class Modeling: Object and class concepts; Link and associations concepts; Generalization and inheritance; A sample class model; Navigation of class models; Practical tips.

UNIT-II **12 Hrs.**

Advanced Class Modeling, State Modeling: Advanced object and class concepts; Association ends; N-ary associations; **Aggregation;** Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived data; Packages; Practical tips.

State Modeling: Events, States, Transitions and Conditions; State diagrams; State diagram behavior; Practical tips. **Advanced State Modeling, Interaction Modeling:** Advanced State Modeling: Nested state diagrams; Nested states; Signal generalization; Concurrency; A sample state model; Relation of class and state models; Practical tips. Interaction Modeling: Use case models; Sequence models; Activity models. Use case relationships; Procedural sequence models; Special constructs for activity models

UNIT-III **12 Hrs.**

Process Overview, System Conception, Domain Analysis

Process Overview: Development stages; Development life cycle. **System Conception:** Devising a system concept; Elaborating a concept; Preparing a problem statement.

Domain Analysis: Overview of analysis; Domain class model; Domain state model; Domain interaction model; Iterating the analysis.

Application Analysis, System Design: Application Analysis: Application interaction model; Application class model; Application state model; Adding operations. **Overview of system design**; Estimating performance; Making a reuse plan; Breaking a system in to sub-systems; Identifying concurrency; Allocation of sub-systems; Management of data storage; Handling global resources; Choosing a software control strategy; Handling boundary conditions; Setting the trade-off priorities; Common architectural styles; Architecture of the ATM system as the example.

UNIT-IV **12 Hrs.**

Class Design, Implementation Modeling, Legacy Systems: Class Design: Overview of class design; Bridging the gap; Realizing use cases; Designing algorithms; Recursing downwards, Refactoring; Design optimization; Reification of behavior; Adjustment of inheritance; Organizing a class design; ATM example. **Implementation Modeling:** Overview of implementation; Fine-tuning classes; Fine-tuning generalizations; Realizing associations; Testing. **Legacy Systems:** Reverse engineering; Building the class models; Building the interaction model; Building the state model; Reverse engineering tips; Wrapping; Maintenance.

Design Patterns, Idioms: What is a pattern and what makes a pattern? Pattern categories; Relationships between patterns; Pattern description. Communication Patterns: Forwarder-Receiver; Client-Dispatcher-Server; Publisher-Subscriber. **Management Patterns:** Command processor; View handler. Idioms: Introduction; what can idioms provide? Idioms and style; Where to find idioms; Counted Pointer example.

REFERENCE BOOKS:

1. Michael Blaha, James Rumbaugh: Object-Oriented Modeling and Design with UML, 2nd Edition, Pearson Education, 2005. (Chapters 1 to 17, 23)
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern-Oriented Software Architecture, A System of Patterns, Volume 1, John Wiley and Sons, 2006. (Chapters 1, 3.5, 3.6, 4)
3. Grady Booch et al: Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson, 2007.
4. Brahma Dathan, Sarnath Ramnath: Object-Oriented Analysis, Design, and Implementation, Universities Press, 2009.
5. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, Wiley-Dreamtech India, 2004.
6. Simon Bennett, Steve McRobb and Ray Farmer: Object-Oriented Systems Analysis and Design Using UML, 2nd Edition, Tata McGraw-Hill, 2002.

CSP 459: OPERATING SYSTEM LAB

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1: Appreciate the advantages of Unix OS.
CO2: Develop and debug, C programs created on UNIX platforms.
CO3: Use and if necessary install standard libraries.
CO4: Developing low-level operating system code.
CO5: Understanding the performance trade-offs in developing high-performance low-level OS code

CSP 460: DATA SCIENCE LAB

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1: Understand the concept and challenge of big data and why existing technology is inadequate to analyze the big data
CO2: Collect, manage, store, query, and analyze various form of big data
CO3: Gain hands-on experience on large-scale analytics tools to solve some open big data problems
CO4: Understand the impact of big data for business decisions and strategy.

CSS 461: MACHINE LEARNING LAB

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1: Understand the implementation procedures for the machine learning algorithms
CO2: Design Java/Python programs for various Learning algorithms.
CO3: Apply appropriate data sets to the Machine Learning algorithms
CO4: Identify and apply Machine Learning algorithms to solve real world problems

CSP 462: DBMS LAB

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1: Design and implement a database schema for a given problem domain.
CO2: Populate and query a database using SQL DDL/DML commands.
CO3: Program in PL/SQL including stored procedures, stored functions, cursors, packages.
CO4: Design and build a GUI application using a 4GL

CSP 463: Natural Language Processing Lab

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1: The students will get acquainted with natural language processing and learn how to apply basic algorithms in this field.

CO2: They will understand the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics, as well as the resources of natural language data - corpora.

CO3: They will also grasp basics of knowledge representation, inference, and relations to the artificial intelligence.

CSP 464: Internet of Things Lab

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1: Identify the requirements for the real world problems.
CO2: Conduct a survey of several available literatures in the preferred field of study.
CO3: Study and enhance software/ hardware skills.
CO4: Demonstrate and build the project successfully by hardware requirements, coding, emulating

OPEN ELECTIVE-I
IN
II SEMESTER

CSE 465 :WEB TECHNOLOGIES

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Employ fundamental computer theory to basic programming techniques.
CO2: Use fundamental skills to maintain web server services required to host a website.
CO3: Select and apply markup languages for processing, identifying, and presenting of information in web pages.
CO4: Use scripting languages and web services to transfer data and add interactive components to web pages.
CO5: Create and manipulate web media objects using editing software.
CO6: Incorporate aesthetics and formal concepts of layout and organization to design websites that effectively communicate using visual elements.

UNIT-I **12 Hrs.**

Web Essentials: Clients, Servers, and Communication. The Internet-Basic Internet Protocols -The World Wide Web-HTTP request message-response message-Web Clients Web Servers-Case Study. Markup Languages: XHTML. An Introduction to HTML History-Versions-Basic XHTML Syntax and Semantics-Some Fundamental HTML Elements-Relative U RLs-Lists-tables-Frames-Forms-XML Creating HTML Documents Case Study. Style Sheets: CSS-Introduction to Cascading Style Sheets-Features-Core Syntax-Style Sheets and HTML Style Rle Cascading and Inheritance-Text Properties-Box Model Normal Flow Box Layout-Beyond the Normal Flow-Other Properties-Case Study. Client- Side Programming: The JavaScript Language-History and Versions Introduction JavaScript in Perspective-Syntax-Variables and Data Types-Statements-Operators-Literals-Functions-Objects-Arrays-Built-in Objects-JavaScript Debuggers.

UNIT-II **12 Hrs.**

Host Objects : Browsers and the DOM-Introduction to the Document Object Model DOM History and Levels-Intrinsic Event Handling-Modifying Element Style-The Document Tree-DOM Event Handling-Accommodating Noncompliant Browsers Properties of window-Case Study. Server-Side Programming: Java Servlets- Architecture -Overview-A Servelet-Generating Dynamic Content-Life Cycle-Parameter Data-Sessions-Cookies URL Rewriting-Other Capabilities-Data Storage Servelets and Concurrency-Case Study- Related Technologies.

UNIT-III **12 Hrs.**

Representing Web Data: XML-Documents and Vocabularies-Versions and Declaration - Namespaces JavaScript and XML: Ajax-DOM based XML processing Event-oriented Parsing: SAX-Transforming XML Documents-Selecting XML Data :XPATH-Template-based Transformations: XSLT-Displaying XML Documents in Browsers-Case Study- Related Technologies. Separating Programming and Presentation: JSP Technology Introduction-JSP and Servlets-Running JSP Applications Basic JSP-JavaBeans Classes and JSP-Tag Libraries and Files-Support for the Model-View-Controller Paradigm-Case Study-Related Technologies.

UNIT-IV **12 Hrs.**

Web Services: JAX-RPC-Concepts-Writing a Java Web Service-Writing a Java Web Service Client-Describing Web Services: WSDL- Representing Data Types: XML Schema-Communicating Object Data: SOAP Related Technologies-Software Installation-Storing Java Objects as Files-Databases and Java Servlets.

REFERENCE BOOKS:

1. Jeffrey C. Jackson, "Web Technologies--A Computer Science Perspective", Pearson Education, 2006.
2. Robert. W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson Education, 2007.
3. Deitel, Deitel, Goldberg, "Internet & World Wide Web How To Program", Third Edition,

Pearson Education, 2006.
4. Marty Hall and Larry Brown,"Core Web Programming" Second Edition, Volume I and II, Pearson Education, 2001.
5. Bates, "Developing Web Applications", Wiley, 2006.

CSE 466 : LINUX ENVIRONMENT SYSTEMS

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Understanding the basic set of commands and utilities in Linux/UNIX systems.
CO2: To learn to develop software for Linux/UNIX systems.
CO3: To learn the C language and get experience programming in C.
CO4: To learn the important Linux/UNIX library functions and system calls.
CO5: To understand the inner workings of UNIX-like operating systems.
CO6: To obtain a foundation for an advanced course in operating systems.

UNIT-I **12 Hrs.**

Logging In and Logging Out, Anatomy of Linux OS, Directory Structure, /usr Directory, File Types: User datafiles, System data files, Executable files. Naming files and directories, Spawning Processes. **Shell:** Creating User Account, Shell Program, bash shell, Changing shell prompt. **Commands:** Basic Syntax for a command, Exploring the Home Directory, ls, mkdir, rmdir, stat, cat, rm, mv, cp

UNIT-II **12 Hrs.**

Editor: Vi editor. **Hooking up Hardware Devices:** Formatting a Floppy Disk**,** Gathering important system information. Backing Up and restoring the File **System:** Simple Backup, gzip, gunzip, tar. **Printing files:** Print Spool directory, Sending files to Printer.

UNIT-III **12 Hrs.**

Sharing Files with other Users: Maintaining User Accounts, Changing Password, Creating Group Accounts, Granting Access to files, Changing File Ownership, Protecting Files, Making a File Read-Only. Working with Processes**:** Types of processes, ps Command, Creating process, killing process, free command and top utility.

UNIT-IV **12 Hrs.**

Managing Disk Space: df, du commands, Creating Additional Free Disk Space, Locating Unused Files, Setting System Clock. Communication Utilities: who, who am i, finger, mesg, write, wall, talk, Creating a message of the day. X Window System, Graphical User Interfaces: KDE and GNOME Desktop Environment.

REFERENCE BOOKS:

1. Craig and Coletta Witherspoon, SAMS Teach Yourself Linux, First Edition, SAMS Publication, 2007.
2. Richard Petersen, Red Hat Linux - The Complete Reference Second Edition McGraw-Hill, 2002

III SEMESTER

III SEMESTER M.Sc. Computer Science								
Subject Code	Course	Theory Hours / Week	Practical Hours / Week	Duration of exams (Hrs)	Marks & Credits			
					IA	Exam	Total	Credits
HARD CORE								
CSH501	**.NET Technology**	4L	-	3	30	70	100	4
CSH502	**Computer Graphics and Multimedia**	4L	-	3	30	70	100	4
CSH503	**Software Engineering**	4L	-	3	30	70	100	4
SOFT CORE								
CSS504 **CSS505** **CSS506** **CSS507** **CSS508**	**Information Retrieval** **Big Data Analytics** **Android Applications** **Digital Image Processing** **Cloud Computing**	4L	-	3	30	70	100	4
CSP509 **CSP510** **CSP511**	**.NET Programming Lab** **Big Data Analytics Lab** **Android Applications Lab**	---	6	3	30	70	100	3
CSP512 **CSP513** **CSP514**	**Computer Graphics Lab** **Image Processing Lab** **Cloud Computing Lab**	---	6	3	30	70	100	3
OPEN ELECTIVE								
CSE515	**Open Source Technologies**	3L		3	30	70	100	3*
CSE516	**Mobile E-Commerce**							
	Total	**19**	**12**	**21**	**210**	**490**	**700**	**22+3***

*** Not included for CGPA.**

CSH501: NET TECHNOLOGY

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: To learn the basics of .net framework and C# language
CO2: To learn c# elements and OOPs concepts
CO3: To learn interface and inheritance concepts in C# language
CO4: To learn fundamentals of window application programming and create a window application
CO5: To develop web applications and learn advanced features of C#
CO6: To develop assemblies and deployment in .Net

UNIT-I **12 Hrs.**

THE PHILOSOPHY OF .NET: Understanding the Previous State of Affairs, The .NET Solution, The Building Block of the .NET Platform (CLR,CTS, and CLS), The Role of the .NET Base Class Libraries, An Overview of .NET Binaries, The Role of the Common Intermediate Language, The Role of .NET Type Metadata, The Role of the Assembly Manifast, Compiling CIL to Platform –Specific Instructions, Understanding the Common Type System, Intrinsic CTS Data Types, Understanding the Common Languages Specification, Understanding the Common Language Runtime A tour of the .NET Namespaces, Increasing Your Namespace Nomenclature, Deploying the .NET Runtime.

UNIT-II **12 Hrs.**

BUILDING C# APPLICATIONS: The Role of the Command Line Complier (csc.exe), Building C # Application using csc.exe Working with csc.exe Response Files, Generating Bug Reports , Remaining C# Compiler Options, The Command Line Debugger (cordbg.exe) Using the, Visual Studio .NET IDE, Other Key Aspects of the VS.NET IDE, C# "Preprocessor:" Directives.

C# LANGUAGE FUNDAMENTALS: The Anatomy of a Basic C# Class, Creating objects: Constructor Basics, The Composition of a C# Application, Default Assignment and Variable Scope, The C# Member Initialization Syntax, Basic Input and Output with the Console Class, Understanding Value Types and Reference Types, The Master Node: System, Object, The System Data Types (and C# Aliases), Converting Between Value Types and Reference Types: Boxing and Unboxing, Defining Program Constants, C# Iteration Constructs, C# Controls Flow Constructs, The Complete Set of C# Operators, Defining Custom Class Methods, Understating Static Methods, Methods Parameter Modifies, Array Manipulation in C #, String Manipulation in C#, C# Enumerations, Defining Structures in C#, Defining Custom Namespaces.

UNIT-III **12 Hrs.**

OBJECT- ORIENTED PROGRAMMING WITH C#: Forms Defining of the C# Class, Definition the "Default Public Interface" of a Type, Recapping the Pillars of OOP, The First Pillars: C#'s Encapsulation Services, Pseudo- Encapsulation: Creating Read-Only Fields, The Second Pillar: C#'s Inheritance Supports, keeping Family Secrets: The "Protected" Keyword, Nested Type Definitions, The Third Pillar: C #'s Polymorphic Support, Casting Between.

EXCEPTIONS AND OBJECT LIFETIME: Ode to Errors, Bugs, and Exceptions, The Role of .NET Exception Handling, Exception Base Class, Throwing a Generic Exception, Catching Exception, CLR System – Level Exception (System. System Exception), Custom Application-Level Exception (System. System Exception), Handling Multiple Exception, The Family Block, the Last Chance Exception Dynamically Identifying Application – and System Level Exception Debugging System Exception Using VS. NET, Understanding Object Lifetime, the

CIT of “new’, The Basics of Garbage Collection,, Finalization a Type, The Finalization Process, Building an Ad Hoc Destruction Method, Garbage Collection Optimizations, The System. GC Type

UNIT-IV **12 Hrs.**

INTERFACES AND COLLECTIONS: Defining Interfaces Using C# Invoking Interface Members at the object Level, Exercising the Shapes Hierarchy, Understanding Explicit Interface Implementation, Interfaces As Polymorphic Agents, Building Interface Hierarchies, Implementation Interfaces Using .NET, understanding the IConvertible Interface.

Callback Interfaces, Delegates, and Events, Advanced Techniques: Understanding Callback Interfaces, Understanding the .NET Delegate Type, Members of System. Multicast Delegate, Understanding Asynchronous Delegates, Understanding (and Using) Events. The Advances Keywords of C#, A Catalog of C# Keywords Building a Custom Indexer, Using C# Indexer from VB .NET. Overloading operators, The Internal Representation of Overloading Operators, interacting with Overload Operator from Overloaded- Operator- Challenged Languages, Creating Custom Conversion Routines, Defining Implicit Conversion Routines.

UNDERSTANDING .NET ASSEMBLES: An Overview of .NET Assembly, Building a Simple File Test Assembly, A C#. Client Application, A Visual Basic .NET Client Application, Cross Language Inheritance, Exploring the CarLibrary’s, Manifest, Exploring the CarLibrary’s Types, Building the Multifile Assembly, Using Assembly, Understanding Private Assemblies, Probing for Private Assemblies (The Basics), Private A Assemblies XML Configurations Files, Probing for Private Assemblies (The Details), Understanding Shared Assembly, Understanding Shared Names, Building a Shared Assembly, Understanding Delay Signing, Installing/Removing Shared Assembly.

REFERENCE BOOKS:

1. Andrew Troelsen, **Pro C# with .NET 3.0** Special Edition, Dream tech Press, India, 2007.
2. E. Balagurusamy, **Programming in C#,** 5th Reprint, Tata McGraw Hill, 2004. (For Programming Examples)
3. Tom Archer, **Inside C#** WP Publishers, 2001.
4. Herbert Schildt**, C#: The Complete Reference**, Tata McGraw Hill, 2004.

\

CSH502: COMPUTER GRAPHICS AND MULTIMEDIA

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: To list the basic concepts used in computer graphics.
CO2: To implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping.
CO3: To describe the importance of viewing and projections.
CO4: To define the fundamentals of animation, virtual reality and its related technologies.
CO5: To understand atypical graphics pipeline
CO6: To design an application with the principles of virtual reality.

UNIT-I **12 Hrs.**

Introduction: Survey of computer Graphics and its applications; Interactive and passive graphics; A graphics system: Video display devices, raster scan and random scan system, The synthetic camera model; The programmer's interface; Graphics architectures; Programmable pipelines; The OpenGL API, Primitives and attributes, Color Models – RGB, YIQ, CMY, HSV; Index color model; Viewing functions; Control functions; Graphics Programming: The Sierpinski gasket

UNIT-II **12 Hrs.**

Interactive Graphics; Input devices: physical input devices and logical input devices, Clients and servers Model; Display lists; Graphics modeling using Display lists, Programming event-driven input; Menus; Building animating interactive models.

Geometry: Scalars, points, and vectors; Three-dimensional primitives; Coordinate systems and frames; Modeling a colored cube; 3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations; Matrix representations and Homogeneous coordinates; Concatenation of transformations; OpenGL transformation matrices.

UNIT-III **12 Hrs.**

Viewing: Classical and computer viewing; Viewing with a computer; Positioning of the camera; Introduction to projections; Projections in OpenGL; Classifications of Projections; Parallel-projection ; Perspective-projection; Deriving Matrices for Parallel and Perspective Projections; Projections and shadows.

Clipping and Rasterization; Clipping; Line-segment clipping Algorithms: Cohen–Sutherland algorithm, Liang–Barsky algorithm; Polygon clipping: Sutherland–Hodgman algorithm; Text Clipping; Rasterization; Line Drawing algorithms : Digital Differential Analyzer(DDA) algorithm, Bresenham's algorithm; Circle Drawing algorithm, Polygon rasterization: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms; Hidden-surface removal: Back face detection, Z-buffer method, Painter's algorithm, scan-line algorithm, BSP-trees, Area sub-division method, Ray tracing.; Antialiasing.

UNIT-IV **12 Hrs.**

An Introduction; Multimedia applications; Multimedia System Architecture; Evolving technologies for Multimedia; Defining objects for Multimedia systems; Multimedia Data interface standards ; Multimedia Databases; Compression & Decompression ; Data & File Format standards; Digital voice and audio; video image and animation ; Full motion video ; Storage and retrieval Technologies; Multimedia Authoring & User Interface; Hypermedia messaging; Mobile Messaging; Virtual Reality.

REFERENCE BOOKS:

1. Edward Angel,, Interactive Computer Graphics A Top-Down Approach with OpenGL 5th Edition, Addison-Wesley, 2008.
2. Prabat K Andleigh and Kiran Thakrar, "Multimedia Systems and Design", PHI, 2003.
3. Donald Hearn and Pauline Baker, Computer Graphics - OpenGL Version 2nd Edition, Pearson Education, 2003.
4. F.S. Hill, Jr. , Computer Graphics Using OpenGL 2nd Edition, Pearson Education, 2001.
5. Ralf Steinmetz, Klara Narstedt, Multimedia Fundamentals: Vol 1-Media Coding and Content Processing 2nd Edition, Pearson Education / PHI, 2003

CSH503: SOFTWARE ENGINEERING

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Understand what software engineering and software process is.
CO2: Understand different activities of Software process.
CO3: Understand the concepts of agile methods.
CO4: Learn the techniques of functional and non functional requirements.
CO5: Become familiar with concepts of detailed and object oriented design.
CO6: Learn concepts of software testing.

UNIT-I **12 Hrs.**

Introduction: Professional Software Development, Software Engineering Ethics. Case Studies. **Software Processes**: Models. Process activities. Coping with Change. The Rational Unified Process.

UNIT-II **12 Hrs.**

Agile Software Development: Agile methods. Plan-driven and agile development. Extreme programming. Agile project management. Scaling agile methods. **Requirements Engineering**: Functional and non-functional requirements. The software Requirements Document. Requirements Specification. Requirements Engineering Processes. Requirements Elicitation and Analysis. Requirements validation. Requirements Management.

UNIT-III **12 Hrs.**

System Models: Context models. Interaction models. Structural models. Behavioural models. Model-driven engineering. **Design and Implementation**: Object-oriented design using the UML. Design patterns. Implementation issues. Open source Development.

UNIT-IV **12 Hrs.**

Software Testing: Development testing, Test-driven development, Release testing, User testing. **Software Evolution**: Evolution processes. Program evolution dynamics. Software maintenance. Legacy system management. **Project Planning**: Software pricing. Plan-driven development. Project scheduling. Agile planning. Estimation techniques. **Quality management:** Software quality. Software standards. Reviews and inspections. Software measurement and metrics.

REFERENCE BOOKS:

1. **Ian Sommerville**: Software Engineering, 9th Edition, Pearson Education, 2012. (Listed topics only from Chapters 1,2,3,4, 5, 7, 8, 9, 23, and 24)
2. **Roger S. Pressman**: Software Engineering-A Practitioners approach, 7th Edition, Tata McGraw Hill.
3. **Pankaj Jalote**: An Integrated Approach to Software Engineering, Wiley India.

SOFT ELECTIVE-II
IN
III SEMESTER

CSS 504: INFORMATION RETRIEVAL SYSTEMS

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Understanding the basics of Information retrieval like what is a corpus, what is precision and recall of an IR system
CO2: Understanding the data structures like Inverted Indices used in Information retrieval systems
CO3: Understanding the basics of web search
CO4: Understanding the different techniques for compression of an index including the dictionary and its posting list
CO5: Understanding the different components of an Information retrieval system
CO6: Developing the ability of develop a complete IR system from scratch.

UNIT-I **12 Hrs.**

Boolean retrieval. The term vocabulary and postings lists. Dictionaries and tolerant retrieval. Index construction. Index compression.

UNIT-II 12 Hrs.

XML retrieval. Probabilistic information retrieval. Language models for information retrieval. Text classification. Vector space classification.
Support vector machines and machine learning on documents. Flat clustering. Hierarchical clustering. Matrix decompositions and latent semantic indexing.

UNIT-III 12 Hrs.

XML retrieval. Probabilistic information retrieval. Language models for information retrieval. Text classification. Vector space classification.
Support vector machines and machine learning on documents. Flat clustering. Hierarchical clustering. Matrix decompositions and latent semantic indexing.

UNIT-IV 12 Hrs.

Web search basics. Web crawling and indexes. Link analysis.

REFERENCE BOOKS:

1. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütz, Introduction to Information Retrieval , Cambridge University Press, 2008.
2. Gerald J. Kowalski, Mark T. Maybury, Information Storage and Retrieval Systems: Theory and Implementation, Springer publication, 2008.
3. Ricardo Baeza-Yates, Modern Information Retrieval, Pearson Education, 2009.
4. David A Grossman and Ophir Frieder, Information Retrieval: Algorithms and Heuristics, 2nd Edition, Springer, 2004.
5. William B Frakes, Ricardo Baeza Yates, Information Retrieval Data Structures and Algorithms, Pearson Education, 1992.

CSS 505: BIG DATA ANALAYTICS

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Work with big data platform and explore the big data analytics techniques business applications.
CO2: Design efficient algorithms for mining the data from large volumes.
CO3: Analyze the HADOOP and Map Reduce technologies associated with big data analytics.
CO4: Explore on Big Data applications Using Pig and Hive.
CO5: Understand the fundamentals of various big data analytics techniques.
CO6: Build a complete business data analytics solution.

UNIT-I **12 Hrs.**

INTRODUCTION TO BIG DATA: Introduction – distributed file system – Big Data and its importance, Four Vs, Drivers for Big data, Big data analytics, Big data applications. Algorithms using map reduce, Matrix-Vector Multiplication by Map Reduce.

UNIT-II **12 Hrs.**

INTRODUCTION HADOOP: Big Data – Apache Hadoop & Hadoop EcoSystem – Moving Data in and out of Hadoop – Understanding inputs and outputs of MapReduce - Data Serialization.

HADOOP ARCHITECTURE: Hadoop Architecture, Hadoop Storage: HDFS, Common Hadoop Shell commands , Anatomy of File Write and Read., NameNode, Secondary NameNode, and DataNode, Hadoop MapReduce paradigm, Map and Reduce tasks, Job, Task trackers - Cluster Setup – SSH & Hadoop Configuration – HDFS Administering –Monitoring & Maintenance.

UNIT-III **12 Hrs.**

HADOOP ECOSYSTEM AND YARN: Hadoop ecosystem components - Schedulers - Fair and Capacity, Hadoop 2.0 New FeaturesNameNode High Availability, HDFS Federation, MRv2, YARN, Running MRv1 in YARN.

UNIT-IV **12 Hrs.**

HIVE AND HIVEQL, HBASE : Introduction to No Query Language, Hive Architecture and Installation, Comparison with Traditional Database, HiveQL - Querying Data - Sorting And Aggregating, Map Reduce Scripts, Joins & Subqueries, HBase conceptsAdvanced Usage, Schema Design, Advance Indexing - PIG, Zookeeper - how it helps in monitoring a cluster, HBase uses Zookeeper and how to Build Applications with Zookeeper.

REFERENCE BOOKS:

1. Boris lublinsky, Kevin t. Smith, Alexey Yakubovich, “Professional Hadoop Solutions”, Wiley, ISBN: 9788126551071, 2015.
2. Chris Eaton, Dirk deroos et al. , “Understanding Big data ”, McGraw Hill, 2012.
3. Tom White, “HADOOP: The definitive Guide” , O Reilly, 2012.
4. Vignesh Prajapati, “Big Data Analytics with R and Haoop”, Packet Publishing 2013.
5. Tom Plunkett, Brian Macdonald et al, “Oracle Big Data Handbook”, Oracle Press, 2014.
6. http://www.bigdatauniversity.com/
7. Jy Liebowitz, “Big Data and Business analytics”,CRC press, 2013.

CSS 506: ANDROID APPLICATION DEVELOPMENTS

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: By the end of the course, student will be able to gain a thorough understanding of Android architecture
CO2: Will be able to write simple GUI applications,
CO3: Use built-in widgets and components, work with the database to store data locally, and much more.
CO4: Acquire the necessary skillsets and experience for professional Android application development by building six top-trending applications during the course.
CO5: Achieve expertise in app development for Android wearable devices

UNIT-I **12 Hrs.**

Introduction to Android Operating System: Android OS design and Features – Android development framework, SDK features, Installing and running applications on Eclipse platform, Creating AVDs, Types of Android applications, Best practices in Android programming, Android tools Android application components – Android Manifest file, Externalizing resources like values, themes, layouts, Menus etc., Resources for different devices and languages, Runtime Configuration Changes Android Application Lifecycle – Activities, Activity lifecycle, activity states, monitoring state changes.

UNIT-II **12 Hrs.**

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

UNIT-III **12 Hrs.**

Intents and Broadcasts: Intent – Using intents to launch Activities, Explicitly starting new Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters, finding and using Intents received within an Activity Notifications – Creating and Displaying notifications, Displaying Toasts

UNIT-IV **12 Hrs.**

Persistent Storage: Files – Using application specific folders and files, creating files, reading data from files, listing contents of a directory Shared Preferences – Creating shared preferences, saving and retrieving data using Shared Preference Database – Introduction to SQLite database, creating and opening a database, creating tables, inserting retrieving and deleting data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update). Advanced Topics: Alarms – Creating and using alarms Using Internet Resources – Connecting to internet resource, using download manager Location Based Services – Finding Current Location and showing location on the Map, updating location.

REFERENCE BOOKS:

1. Reto Meier, Professional Android 4 Application Development, Wiley India, (Wrox) , 2012.
2. James C Sheusi, Android Application Development for Java Programmers, Cengage Learning, 2013
3. Wei-Meng Lee, Beginning Android 4 Application Development, Wiley India (Wrox), 2013

CSS 507: DIGITAL IMAGE PROCESSING

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Understand the need for image transforms different types of image transforms and their properties.
CO2: Develop any image processing application.
CO3: Understand the rapid advances in Machine vision.
CO4: Learn different techniques employed for the enhancement of images.
CO5: Learn different causes for image degradation and overview of image restoration techniques.
CO6: Understand the need for image compression and to learn the spatial and frequency domain techniques of image compression.
CO7: Learn different feature extraction techniques for image analysis and recognition

UNIT-I **12 Hrs.**

Digitized image and its properties: Basic concepts, Image digitization, Digital image properties. Image Preprocessing: Image pre-processing; Histogram processing, Enhancement using arithmetic / logic operations, Basics of spatial filtering, Smoothing spatial filters, Sharpening spatial filters. Brightness and geometric transformations, local preprocessing.

UNIT-II **12 Hrs.**

SEGMENTATION: Thresholding, Edge-based segmentation, Region based segmentation, Matching.

UNIT-III **12 Hrs.**

IMAGE ENHANCEMENT: Image enhancement in the frequency domain: Background, Introduction to the Fourier transform and the frequency domain, Smoothing Frequency-Domain filters, Sharpening Frequency Domain filters, Homomorphic filtering.
IMAGE COMPRESSION: Image compression: Fundamentals, Image compression models, Elements of information theory, Error-Free Compression, Lossy compression.

UNIT-IV **12 Hrs.**

SHAPE REPRESENTATION: Region identification, Contour-based shape representation and description, Region based shape representation and description, Shape classes.
MORPHOLOGY: Basic morphological concepts, Morphology principles, Binary dilation and erosion, Gray-scale dilation and erosion, Morphological segmentation and watersheds.

REFERENCE BOOKS:

1. Milan Sonka, Vaclav Hlavac and Roger Boyle, **Image Processing, Analysis and Machine Vision** 2nd Edition, Thomson Learning, 2001.
2. Rafel C Gonzalez and Richard E Woods**, Digital Image Processing,** 2nd Edition, Pearson Education, 2003.
3. Anil K Jain, **Fundamentals of Digital Image Processing** Pearson Education/Prentice-Hall of India Pvt. Ltd., 1997.
4. B. Chanda, D Dutta Majumder**, Digital Image Processing and Analysis** Prentice-Hall India, 2002.

CSS 508: CLOUD COMPUTING

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1: Analyze the Cloud computing setup with it's vulnerabilities and applications using different architectures.
CO2: Design different workflows according to requirements and apply map reduce programming model.
CO3: Apply and design suitable Virtualization concept, Cloud Resource Management and design scheduling algorithms.
CO4: Create combinatorial auctions for cloud resources and design scheduling algorithms for computing clouds
CO5: Assess cloud Storage systems and Cloud security, the risks involved, its impact and develop cloud application
CO6: Broadly educate to know the impact of engineering on legal and societal issues involved in addressing the security issues of cloud computing.

UNIT-I **12 Hrs.**

Principles of Parallel and Distributed Computing, Introduction to cloud computing, Cloud computing Architecture, cloud concepts and technologies, cloud services and platforms, Cloud models, cloud as a service, cloud solutions, cloud offerings, introduction to Hadoop and Mapreduce. Cloud Platforms for Industry, Healthcare and education, Cloud Platforms in the Industry, cloud applications. Virtualization, cloud virtualization technology, deep dive: cloud virtualization, Migrating in to cloud computing, Virtual Machines Provisioning and Virtual Machine Migration Services, On the Management of Virtual Machines for cloud Infrastructure, Comet cloud, T-Systems.

UNIT-II **12 Hrs.**

Cloud computing Applications: Industry, Health, Education, Scientific Applications, Business and Consumer Applications, Understanding Scientific Applications for Cloud Environments, Impact of Cloud computing on the role of corporate IT. Enterprise cloud computing Paradigm, Federated cloud computing Architecture, SLA Management in Cloud Computing, Developing the cloud: cloud application Design.

UNIT-III **12 Hrs**

Python Basics, Python for cloud, cloud application development in python, Cloud Application Development in Python. Programming Google App Engine with Python: A first real cloud Application, Managing Data in the cloud, Google app engine Services for Login Authentication, Optimizing UI and Logic, Making the UI Pretty: Templates and CSS, Getting Interactive. Map Reduce Programming Model and Implementations.

UNIT-IV **12 Hrs.**

Cloud management, Organizational Readiness and change management in the cloud age ,Cloud Security ,Data security in the cloud, Legal Issues in the Cloud , Achieving Production Readiness for the cloud Services

REFERENCE BOOKS:

1. Raj Kumar Buyya , James Broberg, andrzej Goscinski, Cloud Computing, Wiley , 2013
2. Raj Kumar buyya, Christian Vecchiola,selvi,Mastering Cloud Computing, Wiley, 2013.
3. Arshdeep Bahga, Vijay Madisetti, Cloud Computing, University Press, 2014
4. Kumar Saurab, Cloud computing: Wiley India 2011.
5. Mark C. Chu-Carroll, Code in the Cloud, Pragmatic Bookshelf; 1 edition,(Second part of IV UNIT) 2011,

6. K Chandrasekharan, Essentials of cloud computing, CRC Press.
7. John W. Rittinghouse, James Ransome, Cloud Computing, CRC Press.
8. Dave Shackleford, Virtualization Security 2013. SYBEX, Wiley, 2013.
9. Ahson, Cloud computing and Software Services, Wiley.2011.
10. Sosinsky, Cloud Computing Bible, Wiley India, 2012.
11. Dan C. Marinescu-, Cloud Computing, Morgan Kaufmann, 2013,
12. Kai Hwang, Geoffery C. Fox, Jack J. Dongarra, Distributed and Cloud Computing, Elsevier, 2012.
13. Kenneth A. Lambert , B.L. Juneja, Fundamentals of Python, Cengage Delmar Learning India Pvt,2010.

CSP 509: NET Programming Lab

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1: Create user interactive web pages using ASP.Net
CO2: Create simple data binding applications using ADO.Net connectivity.
CO3: Performing Database operations for Windows Form and web applications.
CO4: Ability to design and implement desktop applications
CO5: Ability to design websites

CSP 510: Big Data Analytics Lab

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1: A graduate with a M.Sc. in Big Data Analytics will have the ability to communicate computer science concepts, designs, and solutions effectively and professionally
CO2: This course is aimed to offer training which prepare students to embark on Big Data Analytics careers which is one of the fastest growing technologies.
CO3: They are also provided a very good foundation for further study at PhD level.
CO4: Prepare and equip students for opportunities in ever changing technology with hands-on industrial training.
CO5: Transform the students to become globally competent professionals through internship.

CSP 511: Android Applications Lab

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1. Apply essential Android Programming concepts.

CO2. Develop various Android applications related to layouts & rich user interactive interfaces

CO3. Develop Android applications related to mobile related server-less database like QLITE

CO4. Ability to incorporate security in android applications

CSP 512: **Computer Graphics Lab**

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1. Using OpenGL for Graphics
CO2. Programming User-interface issues
CO3. Concepts of 2D & 3D object representation
CO4. Implementation of various scan & clipping algorithms
CO5. 2D modeling
CO6. Implementation of illumination model for rendering 3D objects
CO7. Visibility detection & 3D viewing
CO8. Implementation of a project based on learned concepts.

CSP 513: Image Processing Lab

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1. Describe basic image related concepts.
CO2. Explain various image enhancement and restoration techniques.
CO3. Describe color image processing, image compression, image segmentation and representation.
CO4. Describe wavelet transforms.

CSP 514: **Cloud Computing Lab**

Hours/Week: 6 **I.A. Marks: 30**
Credits : 3 **Exam. Marks: 70**

Course Outcomes:

CO1. Ability to use current techniques, skills, and tools necessary for computing practice

CO2. Scientific foundation When faced with a technical problem the student should be able to use applied scientific knowledge

CO3. Tools an ability to use the relevant tools necessary for engineering practice.

CO4. Technical design the technical ability to design a prescribed engineering sub-system

CO5. Design assessment the ability to develop and assess alternative system designs based on technical and non-technical criteria

CO6. Critically analyze case studies to derive the best practice model to apply when developing and deploying cloud based applications.

OPEN ELECTIVE-II
IN
III SEMESTER

CSE 513: OPEN SOURCE TECHNOLOGIES

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1. Able to recognize the benefits and features of Open Source Technology.
CO2. Interpret, Contrast and compare open source products among themselves
CO3. Understand and demonstrate Version Control System along with its commands.

UNIT-I **12 Hrs.**

History and Emergence of Open Source Software: The philosophy of OSS, Richard Stallman, The Cathedral and the Bazaar (CatB), commercial software vs OSS, free software vs freeware. Open source development models. Application Programming Interface (API). GNU Project, Free Software Foundation. Community Building: Importance of Communities in Open Source Movement. JBoss Community. Developing blog, group, forum, social network for social purpose.

UNIT-II **12 Hrs.**

Open Standards: National Information Standards Organization (NISO), The Digital Library Federation (DLF). The Dublin Core Metadata Initiative. MARC standards, Resource Description and Access (RDA). Open Archives Initiative. OAI-PMH. Search / Retrieval via URL (SRU), SRW/CQL. Java Platform, Enterprise Edition (Java EE).

UNIT-III **12 Hrs.**

Open Source Licenses: GNU General Public License (GPL) version 2,3, GNU Lesser General Public License (LGPL) version 2.1,3, GNU Affero General Public License (AGPL) version 3, Apache License, Version 2.0, Artistic License 2.0, etc.
Operating System: The Linux operating system and its use both for desktops and as server software.

UNIT-IV **12 Hrs.**

Webserver: Apache HTTP Server and its flavors. WAMP server (Windows, Apache, MySQL, PHP). Open Source MySQL. Apache, MySQL, PHP, JAVA as development platform.
Unit 7: Open Source Software: Category of Open Source Software. OSS for podcasts, RDBMS, online social networks, etc. open source bibliometric software's like pajek, ucinet, etc

REFERENCES:

1) http://directory.fsf.org/GNU/
2) http://www.diglib.org
3) http://www.entirelyopensource.com/
4) http://www.niso.org/`
5) The Dublin Core Metadata Initiative <http://dublincore.org/>
6) MARC standards <http://www.loc.gov/marc/>
7) Resource Description and Access (RDA) <http://www.rdaonline.org/>
8) WAMP server (Windows, Apache, MySQL, PHP) <http://www.wampserver.com/en/>
9) Open Source MySQL <http://www.mysql.com/>
10) Search / Retrieval via URL (SRU) <http://www.loc.gov/standards/sru/>
11) <http://www.loc.gov/standards/>
12) <http://iesr.ac.uk/use/sru/>
13) <http://java.sun.com/javaee/>
14) <http://www.jboss.org/>

CSE 514: MOBILE E-COMMERCE

Hours/Week: 4 **I.A. Marks: 30**
Credits : 4 **Exam. Marks: 70**

Course Outcomes:

CO1. Describe the importance of IT enabled services and challenges.
CO2. Identify strategic IT planning for software development.
CO3. Recognize enterprise IT architecture for Information technology.
CO4. Use of Information Technology so as to enable them for job in sunrise industries.
CO5. Illustrate various IT web services for betterment of knowledge.
CO6. Use their skills to find out various current IT trends in ITES.

UNIT-I **12 Hrs.**

ELECTRONIC COMMERCE: Traditional commerce and E-commerce – Internet and WWW – Role of WWW – Value Chains – Strategic Business And Industry Value Chains – Role of E-commerce. Packet Switched Networks – TCP/IP Protocol Script – Internet Utility Programmes – SGML, HTML And XML – Web Client And Servers – Web Client/Server Architecture – Intranet And Extranets – Web Based Tools For E-commerce – Security. **MOBILE COMMERCE:** Introduction – Infrastructure of M–Commerce – Types Of Mobile Commerce Services – Technologies Of Wireless Business – Benefits And Limitations, Support, Mobile Marketing & Advertisement, Non– Internet Applications In M–Commerce – Wireless/Wired Commerce Comparisons.

UNIT-II **12 Hrs.**

MOBILE COMMERCE: TECHNOLOGY: A Framework For The Study Of Mobile Commerce – NTT Docomo's I-Mode – Wireless Devices For Mobile Commerce – Towards A Classification Framework For Mobile Location Based Services – Wireless Personal And Local Area Networks –The Impact Of Technology Advances On Strategy Formulation In Mobile Communications Networks.

UNIT-III **12 Hrs.**

MOBILE COMMERCE: THEORY AND APPLICATIONS: The Ecology Of Mobile Commerce – The Wireless Application Protocol – Mobile Business Services – Mobile Portal – Factors Influencing The Adoption of Mobile Gaming Services – Mobile Data Technologies And Small Business Adoption And Diffusion – E–commerce in The Automotive Industry – Location– Based Services: Criteria For Adoption And Solution Deployment – The Role of Mobile Advertising In Building A Brand – M–commerce Business Models.

UNIT-IV **12 Hrs.**

BUSINESS– TO– BUSINESS MOBILE E– COMMERCE: Enterprise Enablement – Email and Messaging – Field Force Automation (Insurance, Real Estate, Maintenance, Healthcare) – Field Sales Support (Content Access, Inventory) – Asset Tracking and Maintenance/Management – Remote IT Support – Customer Retention (B2C Services, Financial, Special Deals) – Warehouse Automation – Security.

REFERENCE BOOKS:

1. E.Brian Mennecke, J.Troy Strader, "Mobile Commerce: Technology, Theory and Applications", Idea Group Inc., IRM press, 2003.
2. Ravi Kalakota, B.Andrew Whinston, "Frontiers of Electronic Commerce", Pearson Education, 2003.
3. P. J. Louis, "M-Commerce Crash Course", McGraw- Hill Companies February 2001.
4. Paul May, "Mobile Commerce: Opportunities, Applications, and Technologies Of Wireless Business" Cambridge University Press March 2001.

IV SEMESTER

IV SEMESTER M.Sc. (Computer Science)							
Subject Code	**Subject**	**Practical Hours/ Week**	**Duration of exams (Hrs)**	**Marks & Credits**			
				IA	**Dissertation + Viva-Voce Exam**	**Total**	**Credits**
CSP 551	**Project Work**	**32**	**-**	**120**	**280 (Report :180 Viva-Voce: 100)**	**400**	**16**
Total Marks of I Semester						**700**	**26**
Total Marks of II Semester						**700**	**22+3***
Total Marks of III Semester						**700**	**22+3***
Total Marks of IV Semester						**400**	**16**
Grand Total Credits of all The Four Semesters						**2500**	**86+6***

Preamble:

Project work has been made a part of M.Sc. course to give students exposure to Software development exercises. The primary emphasis of the project work is to understand and gain the knowledge of the principles of software engineering practices. As such, during the development of the project students shall involve themselves in all the stages of the software development life cycle (SDLC) like requirements analysis, systems design, software development/coding, testing and documentation, with an overall emphasis on the development of reliable software systems. Since, the project work spans over the entire final semester, the students shall be advised to take up projects for solving problems of software industry or any research organization or the real life problems suggested by the faculty in-charge of M.Sc. project work in the Institutions. Topic chosen of work must be nontrivial, analytical and scientific /application-oriented. It must involve substantial original work and/or development effort based on the theme. Solved, off-the- shelf and pirated work is not entertained. Any attempt of plagiarism or use of unfair means will result in rejection of the work. All activities of the Project Development must be time-bound and the equal participation of the team members expected throughout the Development process.

Hours/Week: 40 [Major Project **I.A. Marks: 120**
Credits : 16 [Internal Assessment **Exam. Marks: 280**
Project Report Valuation

Course Outcomes:

CO1: Deal with real world data.
CO2: Familiar about real time IT industry environment.
CO3: Experience about applying the knowledge they got until now.
CO4: Build a whole real time working system which will satisfy all customer's needs.
CO5: Use the Systems Analysis Design paradigm to critically analyze a problem.
CO6: Solve the problems (programming networking database and Web design) in the Information Technology environment.
CO7: Function effectively on teams to accomplish a common goal and demonstrate professional behavior.
CO8: Develop IT-oriented security issues and protocols.
CO9: Design and implement a web page.
CO10: Improve communication and business management skills, especially in providing technical support.