RAMAIAH
Institute of Technology

CURRICULUM

for the Academic year 2021 – 2022

COMPUTER SCIENCE AND ENGINEERING

V & VI SEMESTER B.E

RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
Bangalore – 560054.

About the Institute

Dr. M. S. Ramaiah a philanthropist, founded 'Gokula Education Foundation' in 1962 with an objective of serving the society. M S Ramaiah Institute of Technology (MSRIT) was established under the aegis of this foundation in the same year, creating a landmark in technical education in India. MSRIT offers 17 UG programs and 15 PG programs. All these programs are approved by AICTE. All eligible UG and PG programs are accredited by National Board of Accreditation (NBA). The institute is accredited with **'A^+' grade by NAAC in March 2021** for 5 years. University Grants Commission (UGC) & Visvesvaraya Technological University (VTU) have conferred Autonomous Status to MSRIT for both UG and PG Programs since 2007. The institute is a participant to the Technical Education Quality Improvement Program (TEQIP), an initiative of the Government of India. The institute has 380 competent faculty out of which 60% are doctorates. Some of the distinguished features of MSRIT are: State of the art laboratories, individual computing facility for all faculty members, all research departments active with sponsored funded projects and more than 300 scholars pursuing Ph.D. To promote research culture, the institute has established Centre of Excellence for Imaging Technologies, Centre for Advanced Materials Technology, Centre for Antennas and Radio Frequency systems (CARFS), Center for Cyber Physical Systems & Schneider Centre of Excellence. **M S Ramaiah Institute of Technology has obtained "Scimago Institutions Rankings" All India Rank 65 & world ranking 578 for the year 2020.**

The Entrepreneurship Development Cell (EDC) and Section 8 company "Ramaiah Evolute" have been set up on campus to incubate startups. **M S Ramaiah Institute of Technology secured All India Rank 8th for the year 2020 for Atal Ranking of Institutions on Innovation Achievements (ARIIA), by MoE, Govt. of India.** MSRIT has a strong Placement and Training department with a committed team, a good Mentoring/Proctorial system, a fully equipped Sports department, large air-conditioned library with good collection of book volumes and subscription to International and National Journals. The Digital Library subscribes to online e-journals from Elsevier Science Direct, IEEE, Taylor & Francis, Springer Link, etc. MSRIT is a member of DELNET, CMTI and VTU E-Library Consortium. MSRIT has a modern auditorium and several hi-tech conference halls with video conferencing facilities. The institute has excellent hostel facilities for boys and girls. MSRIT Alumni have distinguished themselves by occupying high positions in India and abroad and are in touch with the institute through an active Alumni Association.

As per the National Institutional Ranking Framework (NIRF), MoE, Government of India, M S Ramaiah Institute of Technology has achieved 65th rank among 1143 top Engineering institutions of India for the year 2021 and is 1st amongst the Engineering colleges affiliated to VTU, Karnataka.

About the Department

Year of Establishment	1984
Names of the Programmes offered	UG: B.E. in Computer Science and Engineering PG: M.Tech. in Computer Science and Engineering PG: M.Tech. in Computer Networks and Engineering Ph.D M.Sc.(Engg.) by Research

The Department of Computer Science and Engineering (CSE) has eminent emeritus professors, 15 faculties with the doctorate degree and 15 pursuing the doctoral studies. The faculty has been publishing research papers in refereed journals and in conference proceedings. The department also conducts vocational courses and proficiency courses on fundamental and new programming languages and computer science concepts. These courses are conducted beyond college hours'/summer semester by the faculty of the department. Some of the faculty are involved in institutional level activities and actively involved in interdisciplinary research activities. The department has state of the art laboratories like SAP, IBM Centre of Excellence and CUDA learning center. Technical seminars, workshops and hackathons are conducted regularly for UG & PG students. The department encourages the students to conduct and participate in extra-curricular/sports activities. The alumni network is very active and regular meeting are conducted by the department. The department is accredited by Nation Board of Accreditation (NBA). The department has MoUs with leading IT Industries like NVIDIA, SAP, IBM and HP. The department conducts subjects with more of hands-on sessions and encourages students to take up MOOC based online courses in NPTEL, IITBombayX, Coursera, Udacity and edX.

VISION OF THE INSTITUTE

To be an Institution of International Eminence, renowned for imparting quality technical education, cutting edge research and innovation to meet global socio-economic needs

MISSION OF THE INSTITUTE

MSRIT shall meet the global socio-economic needs through

1. Imparting quality technical education by nurturing a conducive learning environment through continuous improvement and customization
2. Establishing research clusters in emerging areas in collaboration with globally reputed organizations
3. Establishing innovative skills development, techno-entrepreneurial activities, and consultancy for socio-economic needs

QUALITY POLICY

We at M. S. Ramaiah Institute of Technology strive to deliver comprehensive, continually enhanced, global quality technical and management education through an established Quality Management System complemented by the synergistic interaction of the stake holders concerned

VISION OF THE DEPARTMENT

To build a strong learning and research environment in the field of Computer Science and Engineering that promotes innovation towards betterment of the society

MISSION OF THE DEPARTMENT

1. To produce Computer Science graduates who, trained in design and implementation of computational systems through competitive curriculum and research in collaboration with industry and research organizations.
2. To educate students in technology competencies by providing professionally committed faculty and staff.
3. To inculcate strong ethical values, leadership abilities and research capabilities in the minds of students so as to work towards the progress of the society.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs):

A B.E (Computer Science & Engineering) graduate of Ramaiah Institute of Technology should, within three to five years of graduation

PEO1 Pursue a successful career in the field of Computer Science & Engineering or a related field utilizing his/her education and contribute to the profession as an excellent employee, or as an entrepreneur

PEO2 Be aware of the developments in the field of Computer Science & Engineering, continuously enhance their knowledge informally or by pursuing doctoral studies and engage in research and inquiry leading to new innovations and products

PEO3 Be able to work effectively in multidisciplinary and multicultural environments and Be responsible members and leaders of their communities

PROGRAM OUTCOMES (POs):

The Outcomes of the Bachelor of engineering in Computer Science & Engineering Programme are as follows:

Engineering Graduates must be able to:

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs):

PSO1: Understand the principles, architecture and organization of computers, embedded systems, and computer networks.

PSO2: Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems that include both hardware and software

PSO3: Apply software design and development practices to develop software applications in emerging areas such as IoT, Data Analytics, Social Networks, Cloud and High-Performance Computing.

Curriculum Course Credits Distribution

Semester	Humanities & Social Sciences (HSS)	Basic Sciences / Lab (BS)	Engineering Sciences/ Lab (ES)	Professional Courses - Core (Hard core, soft core, Lab) (PC-C)	Professional Courses- Electives (PC-E)	Other Electives (OE)	Project Work/ Internship (PW /IN)	Total credits in a semester
First		9	11					20
Second	2	8	10					20
Third		4	3	18				25
Fourth		7		18				25
Fifth	3			15	3	3		24
Sixth				11	6	3	4	24
Seventh				14	6			20
Eighth							17	17
Total	**5**	**28**	**24**	**76**	**15**	**6**	**21**	**175**

SCHEME OF TEACHING

V SEMESTER

SI. No	Course Code	Course Name	Category	Credits				Contact Hours
				L	T	P	Total	
1.	CS51	Operating Systems	PCC	3	1	0	4	42+28
2.	CS52	Database Systems	PCC	3	1	0	4	42+28
3.	CS53	Artificial Intelligence	PCC	3	0	1	4	42+28
4.	CS54	Entrepreneurship and IPR	HSMC	3	0	0	3	42
5.	CSE55x	Professional Elective-1	PEC	*	*	*	3	--
6.	CSOExx	Open Elective-1	OEC	*	*	*	3	--
7.	CSL56	Advanced Java Programming Laboratory	PCC	0	0	1	1	28
8.	CSL57	Database Systems Laboratory	PCC	0	0	1	1	28
9.	CSL58	Computer Networks Laboratory	PCC	0	0	1	1	28
Total				--	--	--	**24**	--

Note: Minimum of 2 subjects should have a Tutorial component of 1 credit each.

List of Professional Elective-1

SI. No	Course code	Course Name
1.	CSE551	Advanced Computer Networks
2.	CSE552	Robotic Process Automation Design and Development
3.	CSE553	Operation Research
4.	CSE554	Advanced Algorithms
5.	CSE555	System Simulation

*** All electives are 3 credits; The Course Teaching Faculty shall define the split up L:T:P in the Lesson plan**

VI SEMESTER

SI. No	Course Code	Course Name	Category	Credits				Contact Hours
				L	T	P	Total	
1.	CS61	Compiler Design	PCC	3	1	0	4	42+28
2.	CS62	Unix System Programming and Web Technologies	PCC	3	1	0	4	42+28
3.	CSE63X	Professional Elective-2	PEC	*	*	*	3	--
4.	CSE64X	Professional Elective-3	PEC	*	*	*	3	--
5.	CSOExx	Open Elective- 2	OEC	*	*	*	3	--
6.	CS65	Mini Project	PCC	*	*	4	4	--
7.	CSL66	Unix System Programming & Compiler Design Laboratory	PCC	0	0	1	1	28
8.	CSL67	IoT and Embedded Systems Laboratory	PCC	0	0	1	1	28
9.	CSL68	Web Technologies Laboratory	PCC	0	0	1	1	28
Total				--	--	--	**24**	--

Note: Minimum of 2 subjects should have a Tutorial component of 1 credit each.

List of Professional Electives Elective - 2

SI. No	Course Code	Course Name
1.	CSE631	Data Mining and Machine Learning
2.	CSE632	Software Defined Networks
3.	CSE633	Object Oriented Modelling and Design
4.	CSE634	Natural Language Processing
5.	CSE635	Advanced DBMS

Elective – 3

SI. No	Course Code	Course Name
1.	CSE641	Software Architecture & Design Patterns
2.	CSE642	Wireless Sensor Networks
3.	CSE643	Digital Forensics
4.	CSE644	Computer Graphics and Visualization
5.	CSE645	Information Retrieval

***All electives are 3 credits; The Course Teaching Faculty shall define the split up L:T:P in the Lesson plan**

List of Open Elective Courses

Sl. No	Course code	Course Name
1.	CSOE01	Python application Programming
2.	CSOE02	Mobile application Development
3.	CSOE03	Internet of Things
4.	CSOE04	Web Technologies
5.	CSOE05	Object Oriented Programming with C++
6.	CSOE06	Introduction to Data structures and Algorithms using Python
7.	CSOE07	Programming in Java
8.	CSOE08	Operating Systems
9.	CSOE09	Introduction to Artificial Intelligence and Machine Learning
10.	CSOE10	Introduction to Big Data Analytics

The above list of courses shall be offered during 5th and 6th semester for the students of branches other than CSE and ISE Note:

1. Internship of minimum 4 weeks can be carried out by students after completion of 4th semester or after 6th semester or during 8th semester. The final evaluation is done during 8th semester.
2. The 100 activity point report is evaluated during the 8th semester and students needs to submit 5 reports (Each 20 points)

Operating Systems

Course Code: CS51 **Credits: 3:1:0**
Prerequisites: Nil **Contact Hours: 42+28**
Course Coordinator/s: Chandrika Prasad

Course Contents

Unit 1

Introduction to Operating Systems- What operating systems do, Operating System operations, Process management, Memory management, Storage management, Protection and security. **System Structures**- Operating System Services, System calls, Operating System design and implementation, Operating System structure, Virtual machines,

Unit 2

Process Management - Process concept, Process scheduling, Operations on processes, Inter-process communication. **Process Scheduling**- Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple-Processor scheduling. **Process Synchronization-**Background, The Critical section problem, Synchronization hardware, Semaphores, Classical problems of synchronization.

Unit 3

Deadlocks- System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection and recovery from deadlock. **Memory Management Strategies**- Background, Swapping, Contiguous memory allocation, Paging, Structure of page table, Segmentation. **Virtual Memory Management-** Background, Demand paging, Page replacement, Thrashing.

Unit 4

File System- File concept, Access methods, Directory structure, Protection. **Implementing File System**- File system structure, File system implementation, Directory implementation, Allocation methods, Free space management. **Secondary Storage Structures**-Mass storage structures, Disk structure, Disk scheduling.

Unit 5

Dockers-Docker Basics and Architecture, What is containerization, how are containers different from physical machines and VMs, Docker evolution and architecture, Developments in Docker world, Docker tooling, Basic Docker commands. **Docker Networking**- Introduction, Types of Docker networks, Using Networks, Identifying container networks,

Docker Volumes-Managing data in Docker containers with volumes, Volume file systems and basic Docker image file systems, Creating and managing volumes.

Text Book:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating System Principles, 9th edition, Wiley-India, 2012.

Link:

1. URL: https://docs.docker.com/

Reference Books:

1. William Stallings: Operating systems Internals and Design Principles | Ninth Edition, Pearson Education, 2018
2. Remzi H. Arpaci-Dusseau and Andrea C. Arpaci- Dusseau, Operating systems: Three easy pieces, URL: http://pages.cs.wisc.edu/~remzi/OSTEP/
3. James Turnbull: The Docker Book: Containerization is the new virtualization Kindle Edition, Kindle Edition,2014

Course Outcomes (COs):

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

1. Describe operating system operations and operating system structures (PO- 1,3, PSO-1).
2. Assess different scheduling algorithms and concurrency control techniques to provide co-ordination among threads and processes (PO-1,2,3,4, PSO-1).
3. Examine various methods for handling deadlocks and memory management Techniques (PO-1,2,3,4, PSO-1).
4. Identify file systems and recognize its implementation and secondary storage Structure (PO-1,2,3, PSO-1)
5. Illustrate Dockers techniques for various networking and file system applications (PO-1,2,3,4, PSO-1).

Database Systems

Course Code: CS52 **Credits: 3:1:0**
Prerequisites: Nil **Contact Hours: 42+28**
Course Coordinator/s: Dr. Sini Anna Alex

Course Contents

Unit I

Introduction: Characteristics of Database approach, Actors on the Scene, Workers behind the scene, Advantages of using DBMS approach, Data models, schemas and instances, Three-schema architecture and data independence, Database languages and interfaces, the database system environment, Centralized and client-server architectures, Classification of Database Management systems, Entity-Relationship Model: Conceptual Database using high level Conceptual data models for Database Design, A Sample Database Application, Entity types, Entity sets Attributes and Keys Relationship types, Relationship Sets, Roles and Structural Constraints Weak Entity Types.

Unit II

Relational Model and Relational Algebra: Relational Model Concepts, Relational Model Concepts, Relational Model Constraints and Relational Database Schema Update Operations, Transactions and Dealing with Constraint violations, Unary Relational operations, Relational Algebra Operations from Set Theory, Binary Relational Operations, JOIN and DIVISION, Additional Relational Operations, Examples of Queries in Relational Algebra Relational Database Design Using ER- to-Relational Mapping.

Unit III

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic structure of SQL Queries, Additional Basic Operations, Null values, Aggregate Functions, nested Sub queries, Modification of the Database, Join Expressions, Views, Transactions, Integrity Constraints, SQL Data Types and Schemas, Authorization. Database programming issues and techniques, Embedded SQL.

Unit IV

Database Design: Informal Design Guidelines for Relation Schemas, Functional Dependencies, Normal Forms based on Primary Keys, General Definitions of 2nd

and 3rd Normal Forms, Boyce Codd Normal Forms, Multivalued Dependencies and IV Normal Forms, Join Dependencies and V Normal Forms, Inference Rules, Equivalence and Minimal Cover, Properties of Relational Decomposition, Algorithms for relational database schema design.

Unit V

Transaction Management: Transaction Concept, a Simple Transaction Model, Transaction Atomicity and Durability, Serializability, Transaction Isolation and Atomicity, Transaction Isolation Levels, Implementation of Isolation Levels. Concurrency Control: Lock-Based Protocols, Deadlock Handling. Recovery System: Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm.

Text Books:

1. Elmasri and Navathe: Fundamentals of Database Systems, 5th Edition, Addison-Wesley, 2011.
2. Silberschatz, Korth and Sudharshan: Data base System Concepts, 6th Edition, Tata McGraw Hill, 2011.

Reference Book:

1. C.J. Date, A. Kannan, S. Swamynatham: An Introduction to Database Systems, 8th Edition, Pearson education, 2009.

Course Outcomes (COs):

At the end of the course, students should be able to:

1. Design entity-relationship diagrams to represent simple database applications and convert to Relational model (PO-2, 3, 4, 5, PSO-2)
2. Construct relational algebraic expressions for queries using the concepts of relational database theory (PO-1, 2, 4, PSO-2)
3. Formulate using SQL, solutions to a broad range of query and data update problems (PO-2,3,4,5, PSO-2)
4. Apply Normalization to improve database design (PO-1, 2, PSO-2)
5. Interpret the basic issues of transaction processing, concurrency control and recovery techniques (PO-3, PO-4, PSO-2)

Artificial Intelligence

Course Code: CS53 **Credits: 3:0:1**

Contact Hours: 42+28

Prerequisites: Knowledge of any advanced programming language, Algorithms and Data structures, Elementary Discrete Mathematics or similar.

Course Coordinator/s: Dr. Annapurna P Patil and Shreekant Jere

Course Contents

Unit I

Introduction: What is AI? Foundation and History of Artificial Intelligence. **Intelligent Agents:** Agents and Environments, Rationality, The Nature of Environments, The Structure of Agents. **Problem-solving by search:** Problem-Solving Agents, Example Problems, Searching for Solution, Uniformed Search Strategies, Informed Search Strategies, Heuristic Functions. (Chapter 1, 2, 3 of Textbook 1)

Unit II

Logical Agents: Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic, reasoning patterns in propositional Logic, Effective Propositional Model Checking, Agents Based on Propositional Logic. **First-Order Logic**: Representation Revisited, Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic. **Interference in First-order Logic**: Propositional vs. First-Order Inference, Unification and Lifting, Forward chaining, Backward chaining, Resolution. (Chapter 7, 8, 9 of Textbook 1)

Unit III

Planning: Definition, Planning with State-Space Search, Planning Graphs, Other Planning Approaches Analysis. **Uncertainty:** Acting under Uncertainty, Basic Probability Notations, Inference using Full Joint Distributions, Independence, Bayes' Rule and its Use. **Learning from Examples**: Forms of Learning, Supervised Learning, Learning Decision Trees, Artificial Neural Networks, Support Vector Machines, Ensemble Learning. (Chapter 10, Chapter 13, Chapter 18.1,18.2,18.3, 18.7,18.9,18.10 of Textbook 1)

Unit IV

Natural Language Processing: Language Models, Text Classification, Information Retrieval, Information Extraction. **Natural Language communication:** Phrase Structure Grammars, Syntactic Analysis, Augmented Grammars and Semantic Interpretation, Machine translation, Speech recognition. (Chapter 22, 23 of Textbook 1)

Unit V

Genetic Algorithms: Genetic Algorithms Introduction, Significance of Genetic Operators, Termination Parameters, Niching and Speciation, Evolving Neural Networks, Theoretical Grounding, Ant Algorithms. **Robotics:** Introduction, Hardware, Perception, Planning to Move, Planning Uncertain Movement, Moving, Robotic Software Architecture, Application Domains. **Philosophical Foundations:** Weak and Strong AI, The Ethics and Risks of Developing AI, **AI: The present and Future**. (Chapter 23 of Textbook 2, Chapter 25, 26 ,27 of Text Book 1)

Textbooks:

1. Stuart Russel, Peter Norvig: Artificial Intelligence - A Modern Approach, 3rd Edition, Pearson Education, 2012. (Unit-1,2,3,4,5).
2. Elaine Rich, Kevin Knight, Shivashankar B Nair: Artificial Intelligence, 3rd Edition, Tata McGraw Hill, 2011. (Unit-5).

Reference Books:

1. Peter Jackson, "Introduction to Expert Systems", 3rd Edition, Pearson Education, 2007. (Unit 5).
2. Deepak Khemani "Artificial Intelligence", Tata Mc Graw Hill Education 2013. (unit 3).
3. http://nptel.ac.in

Course Outcomes (COs):

At the end of the course, the student should be able to:

1. Identify problems that are amenable to specific solution by appropriate AI methods. (PO-1,2,3,4,12, PSO-2,3)
2. Utilize various symbolic knowledge representation to specify domains and reasoning tasks of a situated software agent. Use different logical systems for inference over formal domain representations, and trace how an inference algorithm works on a given problem specification. (PO-4,5,9,12, PSO-1,2,3)
3. Formalize a given problem in the language/framework of different AI methods and solve using basic AI algorithms. (PO-1,4,5,6,7, PSO-2,3)
4. Design and carry out an empirical evaluation of different algorithms on a problem formalization and state the conclusions that the evaluation supports. (PO-5,9, PSO-2,3)
5. Communicate scientific knowledge at different levels of abstraction in a variety of research settings. (PO-4,5,9,10,11,12, PSO-2,3)

Entrepreneurship and IPR

Course Code: CS54 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours: 42**
Course Coordinator/s: Dr. Jagadish S Kallimani

Course Contents

Unit I

The Foundations of Entrepreneurship: What's Feeding the Entrepreneurial Fire? The Cultural Diversity of Entrepreneurship. The Power of "Small" Business, the 10 Deadly Mistakes of Entrepreneurship. Putting Failure into Perspective, How to Avoid the Pitfalls. Creativity, Innovation, and Entrepreneurship, Creativity—A Necessity for Survival. Creative Thinking, Barriers to, Creativity How to Enhance Creativity, The Creative Process. Techniques for Improving the Creative Process Protecting Your Ideas.

Unit II

Patents: Introduction, Origin and meaning of the term patent, Objective of a patent law, the legislative provisions regulating patents, principles underlying the patent law in India, patentable invention. Procedure for obtaining patent: Submission of application, Filing provisional and complete specification, Examination of the application, advertisement of the acceptance, opposition, Grant and sealing of patent, Term of the patent, compulsory license. Provisional and complete specification: What is a specification? Kinds of specification, provisional specification, complete specification, Claims, Conditions for amendment.

Unit III

Rights conferred on a patentee: Patent rights, Exception and limitations, Duties of a Patentee. Transfer of patent: Forms of transfer of Patent rights, Assignment, kinds of assignment, License, kinds of license, Rights conferred on a licensee, Transmission of patent by operation of law. Infringement of patents: Construction of claims and infringement, patents held to be infringed, patents held to be not infringed. Action for Infringement: Where a suit is to be instituted, procedure followed in the suit, Onus of establishment infringement, Defense by the defendant, The Relief's, Injunction, Damages or account of profits, patent agents, patent drafting, database searching, and Case studies.

Unit IV

Copy Right: Meaning and characteristics of copy right, Indian copy right law, requirement of copy right, Illustrations copy right in literary work, Musical work, Artistic work, work of architecture, Cinematograph film, sound recording. Author and Ownership of copy right: Ownership of copy right, Contract of service, Contract for service, rights conferred by copy right, terms of copy right, license of copy right. Infringement of copy right: Acts which constitute infringement, general principle, direct and indirect evidence of copying, Acts not constituting infringements, Infringements in literary, dramatic and musical works, Remedies against infringement of copy right, Case studies.

Unit V

Trade Marks: Introduction, Statutory authorities, procedure of registration of trademarks, rights conferred by registration of trademarks, licensing in trade mark, infringement of trade mark and action against infringement. Industrial Design: Introduction, procedure of registration of a design, Piracy of a registered design, Case studies.

Text Books:

1. Management (IV Thomas W. Zimmerer and Norman M. Scarborough: Essentials of Entrepreneurship and Small Business Edition), Prentice Hall, 2004.
2. Dr. B. L. Wadehhra: Intellectual Property Law Handbook, Universal Law Publishing Co.Ltd., 2002.

References Books:

1. Dr. T Ramakrishna: Ownership and Enforcement of Intellectual Property Rights, CIPRA, NSLIU -2005.
2. Intellectual Property Law (Bare Act with short comments), Universal Law Publishing Co. Ltd., 2007.

Course Outcomes (COs):

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

1. Recognize the Basic Principles entrepreneurship (PO-6,8,9, 11, PSO-1).
2. Recognize the procedure get the Patent. (PO-6,8,9, 11, PSO-1).
3. Identify the various rights conferred to Patentee. (PO-6,8,9, 11, PSO-1).
4. Recognize the characteristics and Infringement of Copyright. (PO-6,8,9, 11, PSO-1).
5. Recognize the importance of Trade Marks, Industrial Design and its Infringement. (PO-6,8,9, 11, PSO-1).

Advanced Computer Networks

Course Code: CSE551 **Credits: 3:0:0**
Prerequisites: Data Communication **Contact Hours: 42**
Course Coordinator/s: Sanjeetha R and Meeradevi A K

Course Contents

Unit 1

Next Generation IP: IPv6 Addressing: Representation, The IPv6 Protocol: Packet Format, Extension Header, The ICMPv6 Protocol: Error- Reporting Messages, Information Messages, Neighbor-Discovery Messages, Group Membership Messages, Transition From IPv4 to IPv6: Strategies, Use of IP 21 addresses. Multicast -Multicast Addresses, Multicast Routing (DVMRP, PIM, MSDP), Multiprotocol Label Switching (MPLS), Destination-Based Forwarding, Explicit Routing, Virtual Private Networks and Tunnels, Multipath routing.

Unit 2

Socket Programming with TCP and UDP, TCP NewReno, TCP with Selective Acknowledgments, Forward Acknowledgments, TCP Vegas. High Performance TCP: High-Bandwidth-Delay Products Round-Trip Estimation, Path MTU Discovery, Reducing End-System Overhead: Overhead, CPU Utilization, and Bandwidth, The Role of Application Processing, Sources of Overhead for TCP/IP. Copy Avoidance: Page Remapping, Scatter/Gather I/O, Remote Direct Memory Access and TCP Offload.
Overlay Networks Routing Overlays, Peer-to-Peer Networks, Content Distribution Networks. Network Management: SNMP: Concept, Management Components, SMI, MIB, SNMP format, Messages.

Unit 3

Wireless and Mobile Networks: Introduction, WiFi 802.11 Wireless LANs- The 802.11, Architecture, Mobility in the Same IP Subnet, Advanced Features in 802.11, Mobility Management Principles, Addressing, Routing to a Mobile Node, Mobile IP, Managing Mobility in Cellular Networks, Routing Calls to a Mobile User, Handoffs in GSM. Case studies using simulation environment.

Unit 4

Multimedia Networking: Multimedia Networking Applications: Types of

Multimedia Network Applications, Content Distribution Networks, Voice-overIP: Limitations of the Best-Effort IP Service, Removing Jitter at the Receiver for Audio, Protocols for Real-Time Conversational Applications: RTP, SIP. Case studies related to multimedia applications.

Unit 5

Quality of Service: Traffic Shaping or Policing, Resource Reservation, Admission Control, Integrated Services (Intserv), Differentiated Services (DFFSERV).

Traffic Engineering Basics: Introduction to traffic Engineering, Requirement Definition for Traffic Engineering, Traffic Sizing, Traffic Characteristics, Protocols, Time and Delay Consideration, Connectivity, Availability, Reliability, and Maintainability, Throughput Calculation.

Reference Books:

1. Larry L. Peterson and Bruce S Davie: Computer Networks: A Systems Approach, Fifth Edition, Elsevier, 2011. Tanenbaum: Computer Networks, 4th Ed, Pearson Education/PHI, 2003.
2. William Stallings: Data and Computer Communications, 8 th Edition, Pearson Education, 2012.
3. James F. Kurose and Keith W. Ross: Computer Networking: A Top-Down Approach, 6th edition, Addison-Wesley, 2013.
4. Forouzan: Data Communications and Networking, 5th edition, McGraw Hill Education 2013.

Course Outcomes (COs):

At the end of the course, students should be able to:

1. Describe the concepts of IPv6, Multicast addressing, and MPLS. (PO:1,2,3,4 PSO-1)
2. Illustrate various versions of TCP, overlay networks and SNMP. (PO:1,2,3,4 PSO-1)
3. Identify issues related to wireless networks and mobility in Internet and cellular networks. (PO:1,2,3,4 PSO-1)
4. Describe the concepts of multimedia networking. (PO:1,2,3,4 PSO-1)
5. Illustrate providing Quality of Service and traffic engineering. (PO-1,2,3, 4, PSO-1)

Robotic Process Automation Design and Development

Course Code: CSE552 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours: 42**
Course Coordinator/s: Ganeshayya I Shidaganti

Course Contents

UNIT I

PROGRAMMING BASICS: Introduction to Programming, Data and Data Structure, Algorithms**,** Variables and Arguments**,** Software Application and Software Development Life Cycle (SDLC)**,** Frameworks and Languages **AUTOMATION AND RPA**: History of Automation, Automation and its benefits, Introduction to RPA, Automation vs RPA, Process and Flowchart, RPA Programming Constructs, Robots in RPA, Introduction to Robots, Types of Robots, Benefits and Implementation of RPA

UNIT II

RPA TOOL INTRUDUCTION AND BASICS: RPA Development Life Cycle, How does RPA Work, Challenges in RPA, Variables and Types of Variables, Variables vs. Arguments, Namespaces, and Importing New Namespace.
CONTROL FLOW ACTIVITY: Sequences, Control Flow and its types, Decision control-IF, Switch, IF vs Switch, Loops-Do While, While, For each, Other control flow activities - Delay, Break, Assign, Continue and Parallel

UNIT III

DATA MANIPULATION: Data Manipulation and Its Importance, String Manipulations, Data Table Manipulations, Collection, Its Types and Manipulations.
UI AUTOMATION & SELECTORS: UI interactions, Input actions and Input methods, Containers, Recording & its types, Selectors, Types of Selectors- Full and Partial, Containers and Partial Selectors, Dynamic Selectors

UNIT IV

AUTOMATION CONCEPTS AND TECHNIQUES: Desktop and Web Recording, Extraction and its techniques- Screen scraping, Data scraping and PDF Extraction. Automation Techniques- Workbook and Excel automation (read/write).

EMAIL AUTOMATION: Incoming Email automation - Sending Email automation

Unit V

ERROR AND EXCEPTION HANDLING: Errors, Error handling approach, Try Catch, Retry Scope, Exception Handling, Types of Exceptions, Global Exception Handler, Best Practice for Error Handling

ORCHESTRATOR: Overview, Orchestrator Functionalities, Orchestrator User Interface- Automations, Management and Monitoring

Text Book:

1. Alok Mani Tripathi, Learning Robotic Process Automation, Publisher: Packt Publishing Release Date: March 2018 ISBN: 9781788470940.

Reference Books:

1. Frank Casale (Author), Rebecca Dilla (Author), Heidi Jaynes (Author), Lauren Livingston (Author), Introduction to Robotic Process Automation: a Primer, Institute of Robotic Process Automation.
2. Richard Murdoch, Robotic Process Automation: Guide to Building Software Robots, Automate Repetitive Tasks & Become An RPA Consultant
3. Srikanth Merianda, Robotic Process Automation Tools, Process Automation and their benefits: Understanding RPA and Intelligent Automation
4. https://www.uipath.com/rpa/robotic-process-automation

Course Outcomes (COs):

At the end of the course, students should be able to:

1. Understand Basic Programming concepts and the underlying logic / structure (PO-2, 3, 4, 5, PSO-2)
2. Describe RPA, where it can be applied and how it's implemented. (PO- 1, 2, 4, PSO-2)
3. Describe the different types of variables, Control Flow and data manipulation techniques (PO-1, 2, 4, PSO-2)
4. Identify and understand Image, Text and Data Tables Automation (PO- 2,3,4,5, PSO-2)
5. Describe automation to Email and various types of Exceptions and strategies to handle (PO-3, PO-4, PSO-2)

Operation Research

Course Code: CSE553 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours: 42**
Course Coordinator/s: Dr. Jagadish S Kallimani

Course Contents

Unit I

Introduction, Linear Programming – 1: Introduction: The origin, nature and impact of OR; Defining the problem and gathering data; Formulating a mathematical model; Deriving solutions from the model; Testing the model; Preparing to apply the model; Implementation. Introduction to Linear Programming: Prototype example; the linear programming (LP) model.

Unit II

LP – 2, Simplex Method: Assumptions of LP; Additional examples. The essence of the simplex method; Setting up the simplex method; Algebra of the simplex method; the simplex method in tabular form; Tie breaking in the simplex method, Adapting to other model forms; Post optimality analysis; Computer implementation Foundation of the simplex method. Duality Theory: The revised simplex method, a fundamental insight. The essence of duality theory; Economic interpretation of duality, Primal dual relationship; Adapting to other primal forms.

Unit III

Duality Theory and Sensitivity Analysis, Other Algorithms for LP: The role of duality in sensitive analysis: The essence of sensitivity analysis; Applying sensitivity analysis. The dual simplex method; parametric linear programming: The upper bound technique.

Unit IV

Transportation and Assignment Problems: The transportation problem: A streamlined simplex method for the transportation problem; The assignment problem; A special algorithm for the assignment problem.

Unit V

Game Theory, Decision Analysis: Game Theory: The formulation of two persons, zero sum games; Solving simple games- a prototype example; Games with mixed strategies; Graphical solution procedure; Solving by linear programming,

Extensions. Decision Analysis: A prototype example; Decision making without experimentation; Decision making with experimentation; Decision trees. Metaheuristics: The nature of Metaheuristics, Tabu Search, Simulated Annealing, Genetic Algorithms.

Text Book:

1. Frederick S. Hillier and Gerald J. Lieberman: Introduction to Operations Research: Concepts and Cases, 8th Edition, Tata McGraw Hill, 2005. (Chapters: 1, 2, 3.1 to 3.4, 4.1 to 4.8, 5, 6.1 to 6.7, 7.1 to 7.3, 8, 13, 14, 15.1 to 15.4)

Reference Books:

1. Wayne L. Winston: Operations Research Applications and Algorithms, 4th Edition, Cengage Learning, 2003.
2. Hamdy A Taha: Operations Research: An Introduction, 8th Edition, Pearson Education, 2007.

Course Outcomes (COs):

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

1. Explain the concepts of LPP and formulations. (PO-1,2, PSO-2)
2. Construct problems under simplex methods and its types. (PO-1,2, PSO-2)
3. Identify problems under duality and justify them. (PO-3,5,7, PSO-2)
4. Solve assignment and transportation problems. (PO-4, PSO-2)
5. Illustrate game theory and decision-making problems. (PO-4,11, PSO-2)

Advanced Algorithms

Course Code: CSE554 **Credits: 3:0:0**
Prerequisites: Algorithms **Contact Hours: 42**
Course Coordinator/s: Dr. Jagadish S Kallimani

Course Contents

Unit I

Analysis Techniques: Growth of Functions, Asymptotic notations, Standard notations and common functions, Recurrences and Solution of Recurrence equations – The Substitution method, The recurrence – tree method, The master method, Amortized Analysis: Aggregate, Accounting and Potential Methods.

Unit II

Graph Algorithms: Bellman-Ford Algorithm, Single source shortest paths in a DAG, Johnson's Algorithm for sparse graphs, Maximum bipartite matching. Trees: B-trees, Red- Black trees.
Hashing: General Idea, Hash Function, Separate Chaining, Open addressing, Rehashing, Extendible hashing.

Unit III

Number – Theoretic Algorithms: Elementary notations, GCD, Modular Arithmetic, Solving modular linear equations, The Chinese remainder theorem, Powers of an element, RSA cryptosystem. Heaps: Heaps, Priority Queues, Binomial Heaps, Fibonacci Heaps.

Unit IV

String Matching Algorithms: Naïve string matching, Rabin – Karp algorithm, String matching with finite automata, Knuth-Morris-Pratt algorithm, Boyer-Moore Algorithms.

Unit V

Algorithmic Puzzles: Magic Square, n-queens problem, Glove Selection, Ferrying Soldiers, Jigsaw Puzzle Assembly, A Stack of Fake Coins, Maximum Sum Descent, Hats of Two Colors, Pluses and Minuses, Searching for a Pattern, Locker Doors, Palindrome Counting, Inverting a Coin Triangle, Sorting 5 in 7.

Textbooks:

1. T H Cormen, C E Leiserson, R L Rivest and C Stein: Introduction to Algorithms, 3/e, PHI, 2011.
2. Mark Allen Weiss: Data Structures and Algorithm Analysis in C++, 3rd Edition, Pearson Education, 2011.
3. Anany Levitin and Maria Levitin: Algorithmic Puzzle, Oxford University Press, 2011

Reference Books:

1. Ellis Horowitz, Sartaj Sahni, S Rajasekharan: Fundamentals of Computer Algorithms, University Press, 2007.
2. Alfred V Aho, John E Hopcroft, J D Ullman: The Design and Analysis of Computer Algorithms, Pearson Education, 2011.

Course Outcomes (COs):

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

1. Devise recurrence relations and amortized cost of various operations. (PO- 1,2,4, PSO- 1,2)
2. Illustrate graph algorithms such as Bellman-Ford, Shortest path, and Bipartite matching, B-trees, Red-Black trees and hashing techniques. (PO- 1,2,3, PSO-1,2)
3. Identify the methods for solving modular linear equations, Chinese remainder theorem and RSA cryptosystem, Describe types of heaps such as Binomial and Fibonacci heaps. (PO-1,6,9,10-PSO-1,2)
4. Assess the string-matching algorithms such as Boyer-Moore and Knuth-Morris-Pratt algorithm. (PO-1,2,3,6,9,10, PSO-1,2)
5. Compose mathematical models, objective functions and constraints to solve algorithmic puzzles. (PO-1,2,4,9,10,12, PSO-1,2)

System Simulation

Course Code: CSE555 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours: 42**
Course Coordinator/s: Dr. Divakar Harekal

Course Contents

Unit I

Introduction: When simulation is the appropriate tool and when it is not appropriate, Advantages and disadvantages of Simulation, Areas of application, Systems and system environment components of a system, Discrete and continuous systems, Model of a system, Types of Models, Discrete Event System Simulation, Steps in a Simulation Study. Simulation examples: Simulation of queuing systems, Simulation of inventory systems, Other examples of simulation.

Unit II

General Principles, Simulation Software Concepts in Discrete Event Simulation, The Event Scheduling / Time Advance Algorithm, World Views, Manual simulation Using Event Scheduling, List processing. Simulation in Java, Simulation in GPSS. Queuing Models: Characteristics of Queuing Systems.

Unit III

Random Number Generation, Random Variate Generation-Properties of random numbers, Generation of pseudo random numbers, Techniques for generating random numbers, Tests for Random Numbers. Random Variate Generation: Inverse transform technique, Acceptance Rejection technique, Special properties.

Unit IV

Input Modeling: Data Collection, Identifying the distribution with data, Parameter estimation, Goodness of Fit Tests, Fitting a non-stationary Poisson process, Selecting input models without data, Multivariate and Time Series input models.

Unit V

Output Analysis for a Single Mode: Types of simulations with respect to output analysis, Stochastic nature of output data, Measures of performance and their estimation, Output analysis for terminating simulations, Output analysis for

steady state simulations. Verification and Validation of Simulation Models: Model building, verification and validation, Verification of simulation models, Calibration and validation of models. Optimization via Simulation.

Textbook:

1. Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol: Discrete Event System Simulation, 4th Edition, Pearson Education, 2012.

Reference Books:

1. Lawrence M. Leemis, Stephen K. Park: Discrete –Event Simulation: A First Course, First edition, Pearson / Prentice-Hall, 2006.
2. Averill M. Law: Simulation Modeling and Analysis, 4th Edition, Tata McGraw-Hill, 2011.

Course Outcomes (COs):

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

1. Identify modeling system and types of simulation tools. (PO-1,2 PSO-2)
2. Understand the concepts of scheduling /queuing system using simulation software. (PO-1,2, PSO-2)
3. Test and Analyze random function generation through various transform techniques. (PO-1,2, PSO-2)
4. Analyze the data collection process. (PO-1,2, PSO-2)
5. Interpret the stochastic nature of output data. (PO-1, 2, PSO-2)

Advanced Java Programming Laboratory

Course Code: CSL56 **Credits: 0:0:1**
Prerequisites: OOPS with Java **Contact Hours: 28**
Course Coordinator/s: Pramod C Sunagar

Course Contents:

Sl. No.	Topic
1.	JDBC Connectivity
2.	JDBC Statement Objects
3.	JDBC Result Sets
4.	Transaction Processing
5.	Servlet Life Cycle
6.	doGet and doPost Methods
7.	Servlet with JDBC
8.	Cookies and Session management using Servlet
9.	JSP Elements
10.	JSP Custom Tags
11.	Request Dispatcher
12.	JSP using JDBC
13.	Cookies and Session management using JSP
14.	Java Beans and Socket Programming

Reference Books:

1. Herbert Schildt: Java The Complete Reference, 8th Edition, Tata McGraw Hill, 2013.
2. Y. Daniel Liang: Introduction to JAVA Programming, 7th Edition, Pearson Education, 2012.
3. Jim Keogh: J2EE The Complete Reference, first edition, Tata McGraw Hill, 2011.
4. Ivan Bayross, Sharanam Shah, Cyntiha Bayross and Vishali Shah, Java EE 5 for
5. Beginners, SPD (Sharoff Publishers & Distributors Pvt. Ltd.), 2nd edition August 2008.

Course Outcomes (COs):

This course uses assigned readings, lectures, and homework to enable the students to:

1. Develop java programs to query the database and perform DDL and DML commands. (PO-2, 3, 5. PSO-3)
2. Design and develop the web applications using Servlets. (PO-2, 3, 5 PSO- 3)
3. Design and develop the web applications using JSP. (PO-2, 3, 5. PSO-3)

Database Systems Laboratory

Course Code: CSL57 **Credits: 0:0:1**
Prerequisites: Nil **Contact Hours: 28**
Course Coordinator/s: Ganeshayya I Shidaganti

Course Contents:

1. Introduction to MongoDB and CRUD Operations.
2. MongoDB Usage in Enterprise Applications.
3. Develop an Entity-Relationship (ER) Model and Mapping to Relational Model.
4. Implement SQL Queries using DDL, DML Statements.
5. Build an Application model in Oracle DB using Nested queries, Triggers and Views.
6. Design a Database application for a case study using Visual Basic/Java Script in visual studio /Eclipse Tool.

Reference Books:

1. "Database Management Systems" by Ragh Ramakrishnan, Johanners Gehrke, Second Edition. McGraw-Hill Education.
2. "Fundamentals of Database Systems" by Ramez Elmasri, Shamkant B. Navathe, Fifth Edition, Pearson Publications.
3. "Database System Concepts" by Abraham Silberschatz, Henry F. Korth, sixth Edition, McGraw Hill Education.

Course Outcomes (COs):

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

1. Manage the Core MongoDB Operations (PO-2,5, PSO-1,2)
2. Design an Oracle DB Application Using SQL DDL statements, DML statements and Queries (PO-1,3,4,5, PSO-1,2)
3. Develop a Real time database Application Using IDE of student's choice (PO-1,3,4,5, PSO-1,2)

Computer Networks Laboratory

Course Code: CSL58 **Credits: 0:0:1**
Prerequisites: Data Communication **Contact Hours:28**
Course Coordinator/s: Sanjeetha R and Meeradevi A K

Course Contents:

Note: Student is required to solve one problem from **PART-A** and one problem from **PART-B.** The questions are allotted based on lots. Both questions carry equal marks.

PART – A

Implement the following in C/C++ or Wireshark as suitable.

1. Write a program for error detection using CRC-CCITT (16-bits).
2. Write a program to generate Hamming Code for error detection and correction.
3. Trace Hypertext Transfer Protocol.
4. Trace File Transfer protocol, Trace Transmission control protocol
5. Trace Domain Name Server.
6. Write a client-server program using TCP/IP sockets in which client requests for a file by sending the file name to the server, and the server sends back the contents of the requested file if present.
7. Trace Internet Protocol and Internet Control Message Protocol.
8. Trace Dynamic Host Configuration Protocol.
9. Write a program to implement traffic policing using Leaky bucket algorithm. 10.Write a program to implement traffic policing using Token bucket algorithm.

PART-B

The following experiments shall be conducted using either NS-2/NS3/OMNET++ or any other suitable simulator.

1. Simulate a three nodes point-to-point network with duplex links between them. Set the queue size vary the bandwidth and find the number of packets dropped.
2. Simulate a four node point-to-point network, and connect the links as follows: n0-n2, n1-n2 and n2-n3. Apply TCP agent between n0-n3 and UDP agent between n1-n3. Apply relevant applications over TCP and UDP agents by changing the parameters and determine the number of packets sent by TCP/UDP.

3. Simulate simple Extended Service Set with transmitting nodes in wireless LAN and determine the performance with respect to transmission of packets.
4. Simulate a wireless network, generate traffic and analyze its performance.
5. Simulate a transmission of ping message over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.

Reference Books:

1. James F. Kurose and Keith W. Ross: Computer Networking: A Top-Down Approach, 5th edition, Addison-Wesley, 2009.
2. 2. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff: Unix Network programming, The sockets networking API, Addison-Wesley Professional, 2004
3. Wireshark, "Wireshark", 2021 https://www.wireshark.org/
4. Ns-3,"ns-3manual," 2019 https://www.nsnam.org/docs/manual/html/index.html

Course Outcomes (COs):

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

1. Illustrate networking concepts using programming languages like C/C++/Java / Python. (PO-1, 2, 3,4, 5, *PSO-1*)
2. Use packet sniffing tools like Wireshark to intercept & analyze the packets at different network layers. (PO-1,2, 3, 4,5, *PSO-1*)
3. Use simulators like NS2/NS3. (PO-1, 2, 3, 4, 5, *PSO-1*)

Compiler Design

Course Code: CS61 **Credits: 3:1:0**
Prerequisites: Nil **Contact Hours: 42+28**
Course Coordinator/s: Dr Sini Anna Alex

Course Contents

Unit I

Introduction, Lexical Analysis: Language processors, The Structure of Compilers, Lexical analysis: The Role of Lexical Analyzer, Input Buffering, Specifications of Tokens, recognition of Tokens. Syntax Analysis: Introduction, Writing a Grammar.

Unit II

Parsing: Top-down Parsing, Bottom-up Parsing, Introduction to LR Parsing: Simple LR parser. More Powerful LR Parsers: Canonical parser, LALR parser.

Unit III

Syntax-Directed Definitions: Evaluation order for SDDs, Applications of Syntax-directed translation, Syntax-directed translation schemes. Run-Time Environments: Storage Organization, Stack allocation of space.

Unit IV

Intermediate Code Generation: Variants of syntax trees, Three-address code, Types and declarations, Translation of expressions, Type checking, Control flow, Back patching, and Switch statements, Intermediate code for procedures.

Unit V

Code Generation: Issues in the design of Code Generator, The Target language, Addresses in the target code, Basic blocks and Flow graphs, Optimization of basic blocks, A Simple Code Generator.

Textbook:

1. Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman: Compilers-Principles, Techniques and Tools, 2nd Edition, Pearson education, 2012.

Reference Books:

1. Kenneth C Louden: Compiler Construction - Principles & Practice, First Edition, Brooks/Cole, CENGAGE learning, 1997.
2. Andrew W Appel: Modern Compiler Implementation in C, First Edition, Cambridge University Press, 2010.

Course Outcomes (COs):

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

1. Construct lexical analyzer to recognize inputs using patterns. (PO1,2,4 PSO-1)
2. Devise different types of syntax analyzers using grammars. (PO1,2,4 PSO-1,2)
3. Illustrate syntax-directed translation schemes for grammars. (PO1,3,4,9 PSO-1)
4. Formulate intermediate code generators for programming statements. (PO1,4, PSO-1)
5. Develop assembly language code for the given optimized intermediate codes. (PO1,2,3,4,5 PSO-1,2)

UNIX System Programming and Web Technologies

Course Code: CS62 **Credits: 3:1:0**
Prerequisites: Nil **Contact Hours: 42+28**
Course Coordinator/s: Aparna R

Course Contents

Unit I

Unix Basics & Fundamentals of JavaScript: Unix Basics: Introduction, UNIX Architecture, Files and Directories, File Types, The UNIX File Attributes, Inodes in UNIX, Application Program Interface to Files, Directory Files, Hard and Symbolic Links

HTML5 and Java script Basics: Introduction, Cascading Styles Sheet: Concept of CS, Creating Style Sheet, CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts), CSS Id and Class, Box Model (Border, Padding, Margin properties), CSS Advanced (Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class)

JavaScript syntax, Types of Data and Variables, Operations and calculations, The Document Object, Using Events.

Unit II

Unix File APIs, Process and JavaScript Advanced: File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs Process: The Environment of a UNIX Process: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, UNIX Kernel Support for Processes.

JavaScript Advanced: Scopes and Closures, understand "this" and prototypes, OO concepts as applied to JS and prototypal inheritance, Understanding the meaning of asynchronous. Event loops, Promises.

Unit III

Process Control and Node.js Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid Functions, Race Conditions, exec Functions, Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times, I/O Redirection. Process Relationships: Introduction, Terminal Logins, Network Logins, Process Groups, Sessions, Controlling Terminal.

Introduction to Server-side JS Framework – Node.js: Introduction - What is Node.js, Architecture, Feature of Node JS, Installation and setup - Creating web servers with HTTP (Request & Response), Understand dependence management: npm and package.json File system APIs

Unit IV

Signals and Node.js with MONGODB Signals: The UNIX Kernel Support for Signals, signal, Signal Mask, sigaction, The SIGCHLD Signal and the waitpid Function, Kill, Alarm, Interval Timers. Daemon Processes: Introduction, Daemon Characteristics, Coding Rules, Error Logging, Client-Server Model **CRUD Operations using Node.js:** Event Handling - GET & POST implementation, Use Express.js to create a REST API. Use GET, POST. Connect to NoSQL MongoDB Database using Node.js, Implementation of CRUD operations.

Unit V

Introduction to Client-side JS Framework – Building blocks of React

Create-react-app - Create first React app using this CLI, JSX - Understand what it is and how it's required to create components, Simple functional components, CSS - Load CSS and use it via class Name, props - Passing props to components to make them reusable, Event handling, State - Using class components for storing state (legacy), State via hooks - Using use State hook.

Text Books:

1. Web Application Design and Implementation: Apache 2, PHP5, MySQL, JavaScript, and Linux/UNIX Steven A. Gabarro, December 2006, ©2007, Wiley-IEEE Computer Society Press.
2. Nate Murray, Felipe Coury, Ari Lerner and Carlos Taborda, "ng-book, The Complete Book on Angular 4" September 2016
3. KrasimirTsonev, "Node.js by Example Paperback", May 2015
4. W. Richard Stevens: Advanced Programming in the UNIX Environment, Second Edition, Pearson education, 2011.

Reference Books:

1. Terrence Chan: UNIX System Programming Using C++, First edition, Prentice Hall India, 2011.
2. Kay A Robbins and Steve Robbins: Unix Systems Programming, First Edition, Pearson Education, 2009.
3. Marc J. Rochkind: Advanced UNIX Programming, 2nd Edition, Pearson
4. Web link for Node.js: https://nodejs.org/en/
5. Web link for MongoDB: https://www.mongodb.com/
6. https://reactjs.org/:

Course Outcomes (COs):

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

1. Describe the functions available for file I/O and changing the properties of the file in Unix OS. (PO-3,4, PSO-1)
2. Explain the creation of new process, process accounting and process termination. (PO-1,2,4, PSO-1)
3. Illustrate the basic IPC issues and techniques in UNIX system programming (PO-1,2,4, PSO-3)
4. Explore Node.js and React features and create component based web pages using them (PO3,4, PSO-3)
5. Design Front-end web pages and connect to the Back-end Databases. (PO3,4, PSO-3)

Data Mining and Machine Learning

Course Code: CSE631 **Credits: 3:0:0**
Prerequisites: **Contact Hours: 42**
Course Coordinator/s: Sowmya B J & Dr. Srinidhi H

Course Contents

Unit 1

Data Mining Introduction – Data – Types of Data – Data Mining Functionalities – Classification of Data Mining Systems – Issues –Data Preprocessing. Association Rule Mining Mining Frequent Patterns – Apriori Algorithm Description.

Unit 2

Classification: Basic Concepts, Decision Tree Induction, Bayes Classification Methods, Model Evaluation and Selection, Techniques to Improve Classification Accuracy.

Classification and Prediction: Support Vector Machines – Associative Classification – Lazy Learners – Other Classification Methods – Prediction.

Unit 3

Machine Learning Introduction: Learning, Types of Machine Learning, Types of Machine Learning, Supervised Learning, The Machine Learning Process.

Cluster Analysis: Basic concepts and methods: Cluster Analysis, Partitioning methods, Hierarchical Methods, Evaluation of clustering.

Unit 4

Machine Learning Preliminaries: Terminology - Weight Space, The Curse of Dimensionality; Testing Machine Learning Algorithms – Over-fitting, Training, Testing and Validation Sets, The Confusion Matrix, Accuracy Metrics, ROC Curve, Unbalanced Dataset, Measuring Precision

Turning Data into Probabilities: Minimizing Risk, maximum a posteriori hypothesis; Basic Statistics: Averages, Variance and Covariance, The Gaussian; Bias-Variance Trade-off

Unit 5

Dimensionality Reduction - Unsupervised: Introduction, Subset Selection, PCA (Principal Component Analysis) – Technique, Examples as Numerical. Mining

different types of data: Mining the World Wide Web - Page Rank Algorithm, Text mining, Mining Time Series Data, Ensemble methods-Increasing the Accuracy.

Textbooks:

1. Jiawei Han and Micheline Kamber: Data Mining Concepts and Techniques, Elsevier, 2nd Edition, 2009.
2. Stephen Marsland, "Machine Learning - An AlgorithmicPerspective", Second Edition, CRC Press - Taylor and Francis Group, 2015
3. Ethem Alpaydin, "Introduction to Machine Learning", Second Edition, MITPress, Prentice Hall of India (PHI) Learning Pvt. Ltd. 2010
4. Xindong Wu and Vipin Kumar: The top ten Algorithms in Data Mining, Chapman and Hall/CRC press
5. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, "Introduction to Data Mining", Pearson Education, 2007.
6. DISCOVERING KNOWLEDGE IN DATA, An Introduction to Data Mining, Second Edition, Daniel T. Larose • Chantal D. Larose

Reference Books:

1. K.P. Soman, ShyamDiwakar and V. Aja, "Insight into Data Mining Theory and Practice", Eastern Economy Edition, Prentice Hall of India, 2006.
2. G. K. Gupta, "Introduction to Data Mining with Case Studies", Eastern Economy Edition, Prentice Hall of India, 2006.
3. Christopher Bishop, "Pattern Recognition and Machine Learning", CBS Publishers & Distributors, 2010.
4. Mehryar Mohri, Afshin R, Ameet Talwalkar, "Foundations of Machine Learning", MIT Press, 2012.

Course Outcomes (COs):

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

1. Recognize the preliminary concepts associated with data mining and machine learning such as its need, types, statistical terms and algorithms (PO-2, 4, PSO-3)
2. Differentiate between different classification and prediction techniques. (PO-1,2,3,4, PSO-3)
3. Identify the clustering methods that can be used for a given data set and the need of unsupervised learning using Partitional, Hierarchical

Clustering. (PO-1,2,3,4, PSO-3)

4. Get an introduction to machine learning and its types and appreciate the need for dimensionality reduction and use the same. (PO-1,2,3, PSO-3)
5. Illustrate the use of data mining techniques in various fields like world wide web, time series data and genomic data. (PO-1,2,3,4, PSO-3)

Software Defined Networks

Course Code: CSE632 **Credits: 3:0:0**
Prerequisites: Computer Networks **Contact Hours: 42**
Course Coordinator/s: Sanjeetha R

Course Contents

Unit 1

How SDN Works - Fundamental Characteristics of SDN, SDN Operation, SDN Devices, SDN Controller- SDN controller core modules, SDN controller interfaces, Existing controller implementations, potential issues with the SDN Controller, SDN Applications, Alternate SDN Methods – SDN via APIs, Benefits and Limitations of SDN via APIs, SDN via hypervisor based overlay networks.

The Open Flow Specification – Open Flow Overview – The Open Flow switch, The Open Flow Controller, The Open Flow protocol, The Controller-switch secure channel

Unit 2

Open Flow 1.0 and Open Flow Basics, Open Flow 1.1 Additions, Open Flow 1.2 Additions, Open Flow 1.3 Additions, Open Flow Limitations. Open flow 1.4 additions – Bundles, Eviction and vacancy events, enhanced support for multiple controller, optical port support, and flow table synchronization. OpenFlow 1.5 Additions -Enhanced L4–L7 Support, Pipeline Processing Enhancements, Egress Tables, Fitness for Carrier Use, Bundle Enhancements, Enhanced Support for Multiple Controllers, Enhanced Support for Tunnels, Enhancements to Flow Entry Statistics.

Unit 3

SDN Controllers- Introduction, General Concepts, VMware, Nicira, OpenFlow-Related, Mininet, NOX/POX, Trema, Ryu, Big Switch Networks/Floodlight, Layer 3 Centric, L3VPN, Path Computation Element Server, Plexxi, Plexxi Affinity, Cisco OnePK, Relationship to the Idealized SDN Framework.

Building an SDN Framework- Introduction, The Juniper SDN Framework, IETF SDN Framework(s), SDN(P), ABNO, Open Daylight Controller/Framework, API, High Availability and State Storage, Analytics, Policy

Unit 4

The Journey to Network Functions Virtualization (NFV) Era – NFV Architectural framework- Need for a framework, ETSI framework for NFV, understanding the ETSI framework, A closer look at ETSI's NFV framework, NFV framework summary, Benefits of NFV- Hardware flexibility, faster service life cycle, scalability and elasticity, leveraging existing tools, rapid development and vendor independence, validation of new solutions, amorphous service offering, operational efficiency and agility.

Virtualization of Network Functions: Designing NFV Networks –NFV Design considerations, NFV transformation challenges, Virtualization of Network Infrastructure and Services – NFV for Routing infrastructure, virtualization of network security, virtualization of mobile communication networks

Unit 5

Use Cases for Bandwidth Scheduling, Manipulation, and Calendaring - Introduction, Bandwidth Calendaring, Base Topology and Fundamental Concepts, OpenFlow and PCE Topologies, Example Configuration, OpenFlow Provisioned Example, Enhancing the Controller, Overlay Example Using PCE Provisioning, Expanding Your Reach: Barbarians at the Gate, Big Data and Application Hyper-Virtualization for Instant CSPF, Expanding Topology

Use Cases for Data Center Overlays, Big Data, and Network Function Virtualization- Introduction, Data Center Orchestration, Creating Tenant and Virtual Machine State, Forwarding State, Data-Driven Learning, Control-Plane Signaling, Scaling and Performance Considerations, Puppet (DevOps Solution), Network Function Virtualization - NFV in Mobility, Optimized Big Data

Use Cases for Input Traffic Monitoring, Classification, and Triggered Actions- Introduction, The Firewall, Firewalls as a Service, Network Access Control Replacement, Extending the Use Case with a Virtual Firewall, Feedback and Optimization, Intrusion Detection/Threat Mitigation

Reference Books:

1. Paul Goransson, Chuck Black, Timothy Culver: Software Defined Networks A Comprehensive Approach, Second Edition, Elsevier, 2014.
2. Thomas D.Nadeau & Ken Gray: SDN Software Defined Networks O'Reilly publishers, First edition, 2013.

3. Chayapathi, Rajendra, Syed F. Hassan, and Paresh Shah. Network Functions Virtualization (NFV) with a Touch of SDN: Netw Fun Vir (NFV ePub_1. Addison-Wesley Professional, 2016.

Course Outcomes (COs):

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

1. Describe the fundamental characteristics of SDN. (PO-1,2, 3,4, PSO-3)
2. Differentiate between various OpenFlow specifications. (PO-1,2, 3,4, PSO-3)
3. Illustrate implementation of SDN controllers and building of SDN framework. (PO-1,2, 3,4, PSO-3)
4. Discuss NFV framework and Virtualization of Network Functions. (PO-1,2, 3,4, PSO-3)
5. Illustrate use of SDN and NFV for bandwidth scheduling, data center orchestration and Network Access Control. (PO-1,2, 3,4, PSO-3)

Object Oriented Modeling and Design

Course Code: CSE633 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours: 42**
Course Coordinator/s: Meeradevi A. K

Course Contents

Unit I

Overview of Object Oriented Systems Development: Introduction to Object Oriented Systems- Object Basics- Object Oriented Systems Development Life Cycle Object Oriented Methodologies: Rumbaugh Technique- Booch Methodology- Jacobson Methodology-The Unified Approach
Object Oriented Modeling Language: UML Diagrams- Use Case Diagram-Class Diagram Sequence Diagram –Collaboration Diagram- Activity Diagram-State Machine Diagram Component Diagram-Deployment Diagram-Object Diagram-Package Diagram.

Unit II

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.

Unit III

Advanced Interaction Modeling: Interaction Modeling: Use case models; Sequence models; Activity models. Use case relationships; Procedural sequence models; Special constructs for activity models.
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;

Unit IV

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.

Unit V

Design Patterns – 1: 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.
Design Patterns – 2, IDIOMS: 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. Ali Bahrami, ―Object Oriented Systems Development‖, Irwin McGraw Hill,2nd edition, 2004.
2. Grady Booch, ―Object Oriented Analysis and Design with Applications‖, Pearson Education, 3rd Edition, 2009.
3. Michael Blaha, James Rumbaugh: Object-Oriented Modeling and Design with UML, 2 nd Edition, Pearson Education, 2005. (Chapters 1 to 17, 23)
4. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern-Oriented Software Architecture, A System of Patterns, Volume 1, John Wiley and Sons, 2007.

Course Outcomes (COs):

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

1. Explain the system, component or process as per needs and specifications, Use UML diagrams to design real life problems (PO-1,2,3,4, PSO-3)
2. Understand the purposes, major components and key mechanisms of Class and Object Diagram (PO-1,2,3,4, PSO-3)
3. Analyze advanced interaction modeling process overview and system conception. (PO-1,2,3,4, PSO-3)
4. Analyze class design and implementation modeling (PO-1,2,3,4, PSO-3)
5. Understand patterns, idioms and legacy systems. (PO-1,2,3,4, PSO-3)

Natural Language processing

Course Code: CSE634 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours:42**
Course Coordinator/s: Dr. Jagadish S Kallimani

Course Contents

Unit I

Introduction: Knowledge in Speech and Language Processing, Ambiguity, Models and Algorithms; Language, Thought, and Understanding; The State of the Art and The Near- Term Future; Regular Expressions and Automata; Morphology and Finite-State Transducers: Lexicon-free FSTs: The Porter Stemmer, Human Morphological Processing.

Unit II

N-grams: Counting Words in Corpora, Smoothing, N-grams for Spelling and Pronunciation, Entropy; Word Classes and Part-of-Speech Tagging: Part-of-Speech Tagging, Rule-based Part-of-speech Tagging, Stochastic Part-of-speech Tagging, Transformation-Based Tagging; Context-Free Grammars for English: Constituency, Context-Free Rules and Trees, Sentence- Level Constructions, The Noun Phrase.

Unit III

Parsing with Context-Free Grammars: The Earley Algorithm; Features and Unification: Feature Structures, Unification of Feature Structures, Features Structures in the Grammar, Implementing Unification, Parsing with Unification Constraints; Lexicalized and Probabilistic Parsing: Probabilistic Context-Free Grammars, Problems with PCFGs.

Unit IV

Representing Meaning: First Order Predicate Calculus, Some Linguistically Relevant Concepts, Related Representational Approaches, Alternative Approaches to Meaning; Semantic Analysis: Syntax-Driven Semantic Analysis, Attachments for a Fragment of English; Lexical Semantics: Relations Among Lexemes and Their Senses, WordNet: A Database of Lexical Relations, The Internal Structure of Words.

Unit V

Discourse: Reference Resolution, Text Coherence, Discourse Structure; Generation: Introduction to Language Generation, An Architecture for Generation; Machine Translation: Language Similarities and Differences, The Transfer Metaphor.

Text Book:

1. Daniel Jurafsky and James H Martin, "Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition", Prentice Hall, 2nd Edition, 2008.

Reference Book:

1. Tanveer Siddiqui, U.S. Tiwary, "Natural Language Processing and Information Retrieval", Oxford University Press, 2008.

Course Outcomes (COs):

At the end of the course, the students should be able to:

1. Interpret how speech and language technology relies on formal models to capture knowledge, and language processing deals with subparts of words (morphology). (PO- 1,5,11, PSO-2)
2. Illustrate the way *N*-gram tool is used for spelling and pronunciation processing, and part-of-speech tagging mechanism using various categories. (PO-2,3, 11, PSO-2)
3. Describe feature structures and unification operation which is used to combine them, and probabilistic parsing to capture more syntactic information. (PO-2,11, PSO-2)
4. Outline representations used to bridge the gap from language to commonsense Knowledge (semantic processing), and meanings associated with lexical items. (PO- 1,3,5,11, PSO-2)
5. Emphasize problems that NLP systems face, natural language outputs construction from non-linguistic inputs and machine translation framework approaches. (PO-1,11, PSO-2)

Advanced DBMS

Course Code: CSE635 **Credits: 3:0:0**
Prerequisites: DBMS **Contact Hours: 42**
Course Coordinator/s: Aparna R

Course Contents

Unit I

Database models and overview of Relational data model: Database models: Flat data model, Entity relationship model, Relation model, Record base model, Network model, Hierarchical model, Object oriented data model, Object relation model, Semi structured model, Associative model, Context data model, graph data model, Overview of relational data model, querying relational model with relational algebra, transaction processing and concurrency control.

Unit II

Indexing and Hashing: Basic Concepts, Ordered Indices, B+-Tree Index Files, B+-Tree Extensions, Multiple-Key access, Static Hashing, Dynamic Hashing, Comparison of Ordered Indexing and Hashing, Bitmap Indices, Index Definition in SQL.

Unit III

Query Processing and Optimization: Overview, Measures of Query Cost, Selection Operation, Sorting, Join Operation, Other Operations, Evaluation of Expressions **Query Optimization:** Transformation of Relational Expressions, Estimating Statistics of Expression Results, Choice of Evaluation Plans.

Unit IV

Parallel Databases: Introduction, I/O Parallelism, Inter-query Parallelism, Intra-query Parallelism, Intra-operation Parallelism, Interoperation Parallelism, Query Optimization, Design of Parallel Systems, Parallelism on Multi-core Processors.

Unit V

Distributed Databases: Homogeneous and Heterogeneous Databases, Distributed Data Storage, Distributed Transactions, Commit Protocols, Concurrency Control in Distributed Databases, Availability, Distributed Query

Processing, Heterogeneous Distributed Databases, Cloud-Based Databases, Directory Systems.

Text Books:

1. Database System Concepts" by Abraham Silberschatz , Henry F. Korth, sixth Edition McGraw Hill Education.
2. Fundamentals of Database Systems" by Ramez Elmasri, Shamkant B. Navathe, Fifth Edition, Pearson Publications.

Reference Book:

1. Database Management Systems" by Raghu Ramakrishnan, Johanners Gehrke, Second Edition. McGraw-Hill Education

Course Outcomes (COs):

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

1. Summarise different database models and overview of relation database model (PO-1, PSO-1, 2)
2. Understand the methods of storing, managing and interrogating complex data and the background processes involved in query processing (PO- 1, 3, PSO- 1, 2)
3. Analyze background methods to optimizing the query (PO- 1, 3, PSO- 1, 2)
4. Understand the concept of distributed databases and parallel databases (PO- 1, 3, PSO- 1)
5. Analyze the processing of queries in distributed and parallel databases (PO- 1, 3, PSO-1)

Software Architecture and Design Patterns

Course Code: CSE641 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours: 42**
Course Coordinator/s: Mallegowda M

Course Contents

Unit I

Introduction: The Architecture Business Cycle: Where do architectures come from? Software processes and the architecture business cycle, What makes a "good" architecture? What software architecture is and what it is not, Other points of view, Architectural patterns, reference models and reference architectures, Importance of software architecture, Architectural structures and views.

Unit II

Architectural Styles and Case Studies: Architectural styles, Pipes and filters, Data abstraction and object-oriented organization, Event-based, implicit invocation, Layered systems, Repositories, Interpreters, Process control, Other familiar architectures, Heterogeneous architectures. Case Studies: Keyword in Context, Instrumentation software, Mobile robotics, Cruise control, Three vignettes in mixed style.

Unit III

Quality: Functionality and architecture, Architecture and quality attributes, System quality attributes, Quality attribute scenarios in practice, Other system quality attributes, Business qualities, Architecture qualities. Achieving Quality: Introducing tactics, Availability tactics, Modifiability tactics, Performance tactics, Security tactics, Testability tactics, Usability tactics, Relationship of tactics to architectural patterns, Architectural patterns and styles.

Unit IV

Architectural Patterns: Introduction, From mud to structure: Layers, Pipes and Filters, Blackboard. Distributed Systems: Broker, Interactive Systems: MVC, Presentation- Abstraction- Control. Adaptable Systems: Microkernel, Reflection.

Unit V

Some Design Patterns: Structural decomposition: Whole – Part, Organization of work: Master – Slave, Access Control: Proxy. Designing and Documenting Software Architecture: Architecture in the life cycle, Designing the architecture, Forming the team structure, Creating a skeletal system, Uses of architectural documentation, Views, Choosing the relevant views, Documenting a view, Documentation across views.

Text Books:

1. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, 2nd Edition, Pearson Education, 2011.
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, 2011.
3. Mary Shaw and David Garlan: Software Architecture- Perspectives on an Emerging Discipline, Prentice-Hall of India, 2007.

Reference Books:

1. E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns- Elements of Reusable Object-Oriented Software, 1st Edition, Pearson Education, 2012.
2. Web site for Patterns: http://www.hillside.net/patterns/.

Course Outcomes (COs):

At the end of the course, a student should be able to:

1. Describe the foundational concepts of software architecture and the important principles and techniques of software architecture. (PO-2, 3, 5, PSO-1, 2)
2. Identify the structure, advantages and disadvantages of various architectural choices using case studies. (PO-2, 3, 5, PSO-1, 2)
3. Summarize the need of Software architecture and quality requirements for a software system. (PO-2, 3, 5, 9, PSO-1, 2)
4. Assess different architecture styles and solutions. (PO-2, 3, 5, 9, PSO-1, 2)
5. Apply different architectural views and various design patterns for different software systems. (PO-2, 5, 9, PSO-1, 2)

Wireless Sensor Networks

Course Code: CSE642 **Credits: 3:0:0**
Prerequisites: Computer Networks **Contact Hours: 42**
Course Coordinator/s: Sanjeetha R

Course Contents

Unit I

Introduction: Definition of Wireless Sensor Networks (WSNs), difference between the adhoc and sensor n/ws, challenges for WSN and applications of WSN. Single node architecture: Hardware components, energy consumption in sensor nodes, brief study of operating systems like TinyOS and NesC. Network architecture: Network scenarios, QOS parameters, design principles of WSN and the interfaces.

Unit II

Communication Protocols: Physical layer protocols - Introduction, Wireless channel and communication fundamentals, Physical layer & transceiver design considerations in WSNs. MAC layer protocols Fundamentals of (wireless) MAC protocols, Low duty cycle protocols and wakeup concepts, Contention-based protocols, Schedule-based protocols, The IEEE
802.15.4 MAC protocol, IEEE 802.11 and Bluetooth.

Unit III

Link layer protocols: Fundamentals: Tasks and requirements, Error control, Framing, Link management. Naming and addressing: Fundamentals, Address and name management in wireless sensor networks, Assignment of MAC addresses, Distributed assignment of locally unique addresses, Content-based and geographic addressing.

Unit IV

Time synchronization: Introduction to the time synchronization problem, Protocols based on sender/receiver synchronization, Protocols based on receiver/receiver synchronization. Localization and Positioning: Properties of positioning, possible approaches, Mathematical basics for the lateration problem, Single-hop localization, Positioning in multi-hop environments, Impact of anchor placement.

Unit V

Routing Protocols: The many faces of forwarding and routing, Gossiping and agent-based unicast forwarding, Energy-efficient unicast, Broadcast and multicast, Geographic routing for Mobile nodes.

Text Books:

1. Holger Karl & Andreas Willig, "Protocols and Architectures for Wireless Sensor Networks", John Wiley, 2005.
2. Feng Zhao & Leonidas J. Guibas, “Wireless Sensor Networks- An Information Processing Approach", Elsevier, 2007.

Reference Books:

1. KazemSohraby, Daniel Minoli, & Taieb Znati, “Wireless Sensor Networks- Technology, Protocols, And Applications”, John Wiley, 2007.
2. Anna Hac, “Wireless Sensor Network Designs”, John Wiley, 2003.

Course Outcomes (COs):

At the end of the course, the students should be able to:

1. Explain the hardware and software components of wireless sensor networks. (PO-2, PSO-1, 2).
2. Summarize the fundamentals of MAC protocols. (PO-2, 4, 5, PSO-1, 2)
3. Recognize the important tasks of Link layer addressing and naming schemes in WSNs (PO-1,2, 4, 5, PSO-1).
4. Identify the time synchronization problems and understand the principal design trade- offs for positioning nodes in the network. (PO-1, 2, 5, PSO-1, 2)
5. Compare the different mechanisms available for routing in the network and devise applications (PO-1,2,4, PSO-1, 2)

Digital Forensics

Course Code: CSE643 **Credits: 3:0:0**
Prerequisites: NIL **Contact Hours: 42**
Course Coordinator/s: Pramod C Sunagar

Course Contents

Unit I

Understanding Cyber Crime: Indian IT Act 2008 and amendments, Computer Forensic and Investigations as a Profession, Understanding Computer Forensics. **Understanding Computer Investigations:** Preparing a Computer Investigation, taking a Systematic Approach, Procedures for Corporate High-Tech Investigations, Understanding Data Recovery Workstations and Software.

Unit II

Working with Windows and DOS Systems: Understanding File Systems, Exploring Microsoft File Structures, Examining NTFS Disks, Understanding Whole Disk Encryption, Understanding the Windows Registry, Understanding Microsoft Startup Tasks, Understanding MS-DOS Startup Tasks, and Understanding Virtual Machines.

Unit III

Data Acquisition: Understanding Storage Formats for Digital Evidence, Determining the best Acquisition Method, Contingency Planning for Image Acquisitions, Using Acquisition Tools, Validating Data Acquisitions, Using Remote Network Acquisition Tools. **Computer Forensics Analysis and Validation:** Determining What Data to Collect and Analyze, Validating Forensic Data, Addressing Data-Hiding Techniques, Performing Remote Acquisitions.

Unit IV

Current Computer Forensics Tools: Evaluating Computer Forensic Tool Needs, Computer Forensics Software Tools, Computer Forensics Hardware Tools, Validating and Testing Forensics Software. **Recovering Graphics Files:** Recognizing a Graphics File, Understanding Data Compression, Locating and Recovering Graphics Files, Identifying Unknown File Formats, Understanding Copyright Issues with Graphics.

Unit V

Network Forensics: Network Forensic Overview, Performing Live Acquisitions, Developing Standard Procedures for Network Forensics, Using Network Tools. **E-mail Investigations:** Exploring the Role of E-mail in Investigations, Exploring the Roles of the Client and Server in E-mail, Investigating E-mail Crimes and Violations, Understanding E-mail Servers, Using Specialized E-mail Forensics Tools. Laboratory Lab exercises using forensic software and Case study data.

Textbook:

1. Nelson, Phillips, Frank, Enfinger and Steuart: Computer Forensics and Investigations, Cengage Learning, 2008. (Chapters: 1, 2, 4, 6, 7, 8, 9, 10, 11, 12)

Reference Books:

1. Marjie T. Britz: Computer Forensics and Cyber Crime - An Introduction, 2nd Edition, Pearson Education, 2012.
2. Harish Chander: Cyber Laws and IT Protection, PHI, 2012.
3. http://www.cyberforensics.in/default.aspx

Course Outcomes (COs):

At the end of the course, students should be able to:

1. Recall the Indian IT Act 2008 and its amendments. (PO1,3,4, PSO3)
2. Classify various types of computer crime. (PO1,3,4, PSO3)
3. Apply computer forensic techniques to identify the digital fingerprints associated with criminal activities. (PO1,3,4, PSO3)
4. Analyze hidden information from pictures and other files. (PO1,3,4, PSO3)
5. Apply Network Forensic tools for network forensic and live data forensic analysis. (PO1,3,4, PSO3)

Computer Graphics and Visualization

Course Code: CSE644 **Credits: 3:0:0**
Prerequisites: Data structures **Contact Hours: 42**
Course Coordinator/s: Aparna R

Course Contents

Unit I

Introduction: Applications of computer graphics, A graphics system, Images: Physical and synthetic, Imaging Systems, The synthetic camera model, The programmer's interface, Graphics architectures, Programmable Pipelines Graphics Programming: Programming two- dimensional applications, OpenGL application programming interface, Primitives and attributes, color, viewing, control functions, the gasket program, polygons and recursions, the three dimensional gasket, adding interactions, menus.

Unit II

Geometric Objects and Transformations: Scalars, Points, and Vectors, Three-dimensional Primitives, Coordinate Systems and Frames, Modeling a Colored Cube, Affine Transformations, Rotation, Translation and Scaling, Transformation in Homogeneous Coordinates, Concatenation of Transformations, OpenGL Transformation Matrices, Spinning of cube, Interfaces to three-dimensional applications.

Unit III

Implementation: Basic Implementation Strategies, Four major tasks, Clipping, Line-segment clipping, Polygon clipping, Clipping of other primitives. Clipping in three dimensions, Rasterization: Bresenham's algorithm, Polygon Rasterization, Hidden-surface removal.

Unit IV

Viewing: Classical and computer viewing, Viewing with a Computer, Positioning of the camera, Parallel Projections, Perspective projections, Projections in OpenGL, Hidden-surface removal, Parallel-projection matrices, Perspective-projection matrices, Interactive Mesh Displays, Projections and Shadows.

Unit V

Lighting and Shading: Light and Matter, Light sources, The Phong reflection model, Polygon shading, Approximation of sphere by recursive subdivision, Specifying lighting parameters, Implementing a lighting model/

Text Book:

1. Edward Angel and Dave Shreiner: Interactive Computer Graphics - A Top- Down Approach with Shader-based OpenGL, 6th Edition, Pearson Education, 2011.

Reference Books:

1. Donald Hearn and Pauline Baker: Computer Graphics with OpenGL, 3rd Edition, Pearson Education, 2011.
2. F.S. Hill Jr.: Computer Graphics Using OpenGL, 3rd Edition, Pearson Education, 2009.
3. James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes: Computer Graphics, 2nd Edition, Pearson Education, 2011.

Course Outcomes (COs):

At the end of the course, students should be able to:

1. Understand the components of a graphics system with the building blocks & the overall architecture. (PO-1,5,12 PSO-2)
2. Derive the geometrical transformations used in interactive computer graphics in different coordinate systems and for viewing and projections. (PO-1,5,12 PSO-2)
3. Discuss the different algorithms for clipping and rasterization of lines and polygons, and for hidden surface removal. (PO-2 PSO-2)
4. Illustrate different lighting and shading models. (PO-2,3 PSO-2)
5. Implement 3D computer graphics applications in OpenGL using knowledge of display systems, image synthesis, and interactive control. (PO- 1,3,5,12 PSO-2)

Information Retrieval

Course Code: CSE645 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours: 42**
Course Coordinator/s: Vandana Sudhakar Sardar

Course Contents

UNIT 1

Introduction: Overview, History of IR, **Text Operations:** Document preprocessing, Document Clustering, Text Compression, **Indexing:** Inverted files, **Mathematics for IR:** Set Theory, Mathematical Logic, Probability and Linear algebra, **Classic IR Models:** Boolean Model, Vector space model: tf-idf weighing, Probabilistic Model. **Language models for IR:** The language model and the query likelihood model.

UNIT 2

Evaluation Measures: Precision, Recall, Alternative Measures, **Reference Collections:** TREC, **Relevance Feedback and Query Expansion**, **Text Classification:** The text classification problem, **Flat clustering:** Clustering in information retrieval, Problem Statement, **Hierarchical clustering:** Hierarchical agglomerative clustering, Single-link and Complete-link clustering.

UNIT 3

String Matching algorithms: Knuth Morris Pratt and Rabin Karp, **Stemming algorithm:** Porter, **Map reduce algorithms:** tf- idf calculation and indexing**, Classification:** Naive Bayes algorithm, **Clustering:** k-means algorithm. **Machine learning Algorithms:** Machine- learned scoring, Result ranking by machine learning

UNIT 4

Web search basics: web characteristics**, Architecture of web search engine, Web Crawling and Indexing:** Overview, Crawling, Distributing Indexes, Connectivity Servers. **Link analysis:** Web as a graph**,** Page Rank, Hubs and Authorities.

UNIT 5

XML Retrieval: Basic XML Concepts, Challenges in XML retrieval, a vector space model for XML retrieval, **Introduction to Semantic Web:** Purpose, Semantic Web Stack, RDF, RDFS, Ontology, Web ontology language (OWL) and ontology tools**.**

Text Books:

1. Ricardo Baeza-Yates, Berthier Ribeiro-Neto: Modern Information Retrieval, Pearson Education, 1999.
2. Introduction to Information Retrieval. C.D. Manning, P. Raghavan, H. Schütze. Cambridge UP, 2008.

Reference Books:

1. William B Frakes, Ricardo Baeza Yates: Information Retrieval Data Structures and Algorithms, PH PTR, 1992.
2. David A Grossman, Ophir Frieder: Information Retrieval Algorithms and Heuristics, 2e, Springer, 2004
3. Mathematics for Classical Information Retrieval: Roots and Applications: Dariush Alimohammadi,
4. http://www.dcc.fc.up.pt/~zp/aulas/1213/pde/geral/bibliografia/MIT.Press.A. Semantic. Web.Primer.eBook-TLFeBOOK.pdf

Course Outcomes (COs):

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

1. Describe text operations and various information retrieval models. (PO-1,2, PSO-2)
2. Evaluate an IR system using various evaluation measures. (PO-1,2,4, PSO- 2)
3. Apply various algorithms such as string matching, map reduce, classification and clustering. (PO-1,2,3,4,5, PSO-3)
4. Design web search engine, web crawling and link analysis techniques. (PO- 1,2, 3,4, PSO-3)
5. Explain XML Retrieval and various semantic web technologies. (PO-1, 3, 5, PSO-3)

Mini Project

Course Code: CS65 **Credits: 4**

Course Coordinator/s: Sowmya B J & Dr. Geetha J

Guidelines:

As a part of Mini Project, students must carry out the following activities:

1. Students should form a group to carry out their project. The minimum group size is 3 and maximum group size is 4.
2. The groups will be attached to one Internal Guide by the Department.
3. Students can carry out their project in-house once guide approves the topic.
4. Identify the problem statement based on the current state of Art and trends in the area of interest
5. Based on the survey, identify the project requirements and do feasibility study.
6. Identify and draw a system level architecture by showing subsystems and their input/output need.
7. Implement the programs using step by step for each module.
8. Integrate and examine the implementation and test the project scope and the requirements.
9. Prepare the demonstration of the Project documents and the presentation.
10. The evaluation is based on presentation and report.

Assessment:

- CIE - Minimum 2 reviews of the project + any other relevant components (for 50 Marks)
- SEE - Project Demonstration + any other component as decided by Internal Examination (for 50 Marks)
- The evaluation will be done by the internal guide and a co-examiner twice during the semester.

- Mid-semester evaluation: Students must do a group presentation and produce documents of system requirements, and system design (during 6th week).
- Final Evaluation: At the End of the semester students must do a group presentation, demonstrate the project work and submit the complete report. (During 13th week).

Course Outcomes (COs):

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

1. Formulate a real world problem and develop its requirements. (PO-1,2,3,4,6,11,12, PSO-1,2,3)
2. Develop a design solution for a set of requirements. (PO-1,2,3,4,5,11,12, PSO-1,2,3)
3. Test and validate the conformance of the developed prototype against the original requirements of the problem. (PO-1,2,3,4,5,11,12, PSO-1,2,3)
4. Appraise the Contribution of an individual or in a team in development of the project. (PO-8,9,11,12)
5. Develop effective communication skills for presentation of project related activities (PO-8,9, 10, 11,12)

UNIX System Programming and Compiler Design Laboratory

Course Code: CSL66 **Credits: 0:0:1**
Prerequisites: C programming **Contact Hours: 28**
Course Coordinator/s: Dr. Parkavi A and Aparna R

Course Contents

Part A

1. Basic file I/O functions & properties of a file.
2. File Types, File access permission and File links.
3. Creating the process and process accounting.
4. Feature provided by different signal implementation.
5. Coding rules and Characteristics of Daemon Process

Part B

1. Tokenization of input
2. Validating the syntax of the input
3. Performing syntax directed translation
4. Verification of semantic
5. Generation of intermediate code
6. Optimization of code
7. Generation of assembly language code

Text Books:

1. W. Richard Stevens: Advanced Programming in the UNIX Environment, Second Edition, Pearson education, 2011.
2. Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman:Compilers-Principles, Techniques and Tools, 2nd Edition, Addison-Wesley, 2007.

Reference Books:

1. Terrence Chan: UNIX System Programming Using C++, First edition, Prentice Hall India, 2011.
2. Kay A Robbins and Steve Robbins: Unix Systems Programming, First Edition, Pearson Education, 2009.
3. Marc J. Rochkind: Advanced UNIX Programming, 2nd Edition, Pearson
4. Kenneth C Louden: Compiler Construction - Principles & Practice,

5. Brooks/Cole, CENGAGE learning, 1997.
6. Andrew W Appel: Modern Compiler Implementation in C, Cambridge University Press, 1999.

Course Outcomes (COs):

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

1. Demonstrate the advanced Unix operating system concepts & analyze the different concepts of tokens in a compiler. (PO 1,2,3,4, PSO 2)
2. Create programs for code generation & code optimization utilizing the special commands in Unix. (PO 1,2,3,4, PSO 2)
3. Apply the advanced Unix programming to different compiler sorting tools in different phases. (PO 1,2,3,4, PSO-2)

IoT and Embedded Systems Laboratory

Course Code: CSL67 **Credits: 0:0:1**
Prerequisites: Microprocessors/Microcontrollers **Contact Hours: 28**
Course Coordinator/s: Dr. Divakar Harekal

Course Contents:

1. Study of simple GPIO programs to use the ports.
2. Study useof interrupts and peripherals like LCD 16x2.
3. Study of ADC programs Keyboard, seven segments.
4. Study of ADC programs and graphical LCD 128x64.
5. Study of UART programming.
6. Study of enabling the I2C with LCD 16x2.
7. Study of enabling the I2C.
8. Study use of PWM with DC motor/ servo motor.
9. Study of NodeMCU wireless features. 10.Use of SPI with LCD graphical 128X64.

Textbooks:

1. Wayne Wolf "Computers as Components Principles of Embedded Computer System Design", Second Edition, Elsevier, 2008.
2. Joseph Yiu, "The Definitive Guide to the ARM Cortex-M0 ", 1st edition, Newness – an imprint of Elsevier, 2011.
3. Lyla B. Das, "Embedded Systems an integrated approach "1st edition, Pearson, 2013.

Course Outcomes (COs):

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

1. Investigate a variety of devices & technologies in IoT systems. PO1,3,4, PSO3
2. Design & implement programs using various ICS & micro controllers for IoT & embedded systems. PO1,3,4, PSO3
3. Understand the characteristics of real time by implementing programs to interface memory, I/O with processor. PO1,3,4, PSO3

Web Technologies Laboratory

Course Code: CSL68	**Credits: 0:0:1**
Prerequisites: Nil	**Contact Hours: 28**
Course Coordinator/s: Dr. Geetha J	

Course Contents:

There shall be a minimum of 2 exercises conducted on each of the following topics:

1. HTML5
2. Java Script
3. Advanced java Script
4. Inheritance In java Script
5. Scopes and Closures
6. Develop a Server side programming using java Script : Node.Js
7. Implement CRUD Operation using MongoDB and Node.Js
8. Develop an Client side Application using React Js
9. Implement MERN Stack.

Reference Books:

1. Web Application Design and Implementation: Apache 2, PHP5, MySQL, JavaScript, and Linux/UNIX Steven A. Gabarro, December 2006, ©2007, Wiley-IEEE Computer Society Press.
2. Nate Murray, Felipe Coury, Ari Lerner and Carlos Taborda, "ng-book, The Complete Book on Angular 4" September 2016 3. KrasimirTsonev, "Node.js by Example Paperback", May 2015.
3. Web link for Angular4.0: https://angular.io/
4. Web link for Node.js : https://nodejs.org/en/
5. Web link for MongoDB: https://www.mongodb.com

Course Outcomes (COs):

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

1. Develop web pages with various media contents using HTML5. (PO- 1,2,3 PSO-2)
2. Create a robust Client side validation with java script. (PO-1,2,3 PSO-2)
3. Design dynamic data-driven Web sites using MongoDB and Node.js (PO- 1,2,3,5 PSO-2)

Open Elective Courses

Python Application Programming

Course Code: CSOE01 **Credits: 3**
Prerequisites: Nil **Contact Hours: 42**
Course Coordinator/s: Hanumantharaju R and Pradeep kumar D

Course Contents

Unit I

The way of the program: The Python programming language, What is a program?, What is debugging?, Syntax errors, Runtime errors, Semantic errors, Experimental debugging.

Variables, Expressions and Statements: Values and data types, Variables, Variable names and keywords, Conditional Statements, Evaluating expressions, Operators and operands, Type converter functions, Order of operations, Operations on strings, Input, Composition, The modulus operator. **Iteration:** Assignment, Updating variables, The for loop, The while statement, The Collatz 3n + 1 sequence, Nested Loops for Nested Data. **Strings:** Working with strings as single things, Working with the parts of a string, Length, Traversal and the for loop, Slices, String comparison, Strings are immutable, The in and not in operators, A find function, Looping and counting, Optional parameters, The built-in find method, The split method, Cleaning up your strings, The string format method.

Unit II

Tuples: Tuples are used for grouping data, Tuple assignment, Tuples as return values, Composability of Data Structures.

Lists: List values, Accessing elements, List length, List membership, List operations, List slices, Lists are mutable, List deletion, Objects and references, Aliasing, Cloning lists, Lists and for loops, List parameters, List methods, Pure functions and modifiers, Functions that produce lists, Strings and lists, list and range, Nested lists, Matrices.

Dictionaries: Dictionary operations, dictionary methods, aliasing and copying.

Functions: Functions with arguments and return values, Lambda Functions.

Unit III

Modules: Random numbers, The time module, The math module, Creating your own modules, Namespaces, Scope and lookup rules, Attributes and the dot

operator. **Files:** About files, Writing our first file, Reading a file line-at-a-time, Turning a file into a list of lines, Reading the whole file at once, Working with binary files, Directories, fetching something from the web.
Algorithms: Linear search, Binary search, merging two sorted lists.

Unit IV

Object oriented programming: Classes and Objects — The Basics, Attributes, Adding methods to our class, Instances as arguments and parameters, Converting an instance to a string, Instances as return values, Objects are mutable, Sameness, Copying.
Exceptions: Catching exceptions, Raising our own exceptions, The finally clause of the try statement
Inheritance: Polymorphism, Generalization, Pure functions, Operator Overloading.

Unit V

GUI: Creating Graphical User Interfaces, Using Module Tkinter, Building a Basic GUI, Models, Views, and Controllers, Customizing the Visual Style, Few More Widgets.
Databases: Overview, Creating and Populating, Retrieving Data, Updating and Deleting, Using NULL for Missing Data, Using Joins to Combine Tables, Keys and Constraints, Advanced Features, Developing GUI Application.

Text Books:

1. Downey, A., Elkner, J., & Meyers, C. (2002). How to think like a computer scientist: learning with python. Green Tea Press, Wellesley, Massachusetts.
2. Campbell, J., Gries, P., Montojo, J., & Wilson, G. (2013). Practical programming: an introduction to computer science using Python. Pragmatic Bookshelf, Second Edition.

Course Outcomes (CO's):

At the end of the course, student should be able to:

1. Design and implement Python programs utilizing arithmetic expressions, repetition, file Input and Output.
2. Define and demonstrate the use of the built-in data structures in Python.
3. Employ control structures, functions, and arrays to create Python programs.
4. Understand the concepts of object-oriented programming as used in Python.
5. Define and demonstrate the use of GUI and databases using Python.

Mobile Application Development

Course Code: CSOE02 **Credits: 3:0:0**
Prerequisites: OOPs and Java **Contact Hours:42**
Course Coordinator/s: Pramod C Sunagar

Course Contents

Unit I

Build your first app: Introduction to Android, Create Your First Android App, Layouts, Views and Resources, Text and Scrolling Views, Resources to Help You Learn. **Activities:** Understanding Activities and Intents, The Activity Lifecycle and Managing State, Activities and Implicit Intents. **Testing, debugging, and using support libraries:** The Android Studio Debugger, Testing your App, The Android Support Library. (Lesson 1, 2 and 3)

Unit II

User interaction: User Input Controls, Menus, Screen Navigation, Recycler View. **Delightful user experience:** Drawables, Styles, and Themes, Material Design, Providing Resources for Adaptive Layouts. **Testing your UI:** Testing the User Interface. (Lesson 4, 5 and 6)

Unit III

Background Tasks: Async Task and Async Task Loader, Connect to the Internet, Broadcast Receivers, Services. **Triggering, scheduling and optimizing background tasks:** Notifications, Scheduling Alarms, Transferring Data Efficiently. (Lesson 7 and 8)

Unit IV

Preferences and Settings: Storing Data, Shared Preferences, App Settings. **Storing data using SQLite:** SQLite Primer, SQLite Database. **Sharing data with content providers:** Share Data Through Content Providers. (Lesson 9, 10 and 11)

Unit V

Permissions, Performance and Security: Permissions, Performance and Security. **Firebase and AdMob:** Firebase and AdMob. **Publish:** Publish. (Lesson 13, 14 and 15)

Text Books:

1. Google Developer Training, "Android Developer Fundamentals Course Concept Reference", Google Developer Training Team, 2017. https://www.gitbook.com/book/google-developer-training/android-developer fundamentals-course-concepts/details (Download pdf file from the above link).

Reference Books:

1. Erik Hellman, "Android Programming – Pushing the Limits", 1st Edition, Wiley India Pvt Ltd, 2014.
2. Dawn Griffiths and David Griffiths, "Head First Android Development", 1st Edition, O'Reilly SPD Publishers, 2015.
3. J F DiMarzio, "Beginning Android Programming with Android Studio", 4th Edition, Wiley India Pvt Ltd, 2016. ISBN-13: 978-8126565580
4. Anubhav Pradhan, Anil V Deshpande, "Composing Mobile Apps" using Android, Wiley 2014, ISBN: 978-81-265-4660-2

Course Outcomes (COs):

At the end of the course, students should be able to:

1. Understand the android OS and fundamental concepts in android programming.
2. Demonstrate various components, layouts and views in creating android applications.
3. Design applications to save, or store, data using SQLite and Content Providers.
4. Demonstrate the working of long running tasks in the background using Services.
5. Demonstrates how to write applications using location based services and to publish the android applications.

Internet of Things (IoT)

Course Code: CSOE03 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours:42**
Course Coordinator/s: Dr. Divakar Harekal

Course Contents

Unit I

Introduction to IoT: What is IoT?, IOT terms and Basic Definitions, Disambiguation of IoT vs IoE vs M2M vs Others, Characteristics of IoT. **Wireless Sensor Networks:** Potential Applications, WSN System Architecture, WSN Network Topologies, Components of a WSN Node.

Architecture of IoT systems: Things in IoT, Applications of IoT and IoT Reference model.

Self-Study: IoT Ecosystem, Enabling Technologies in IoT, Marketplace and Vision of IoT.

Unit II

Hardware aspects of IoT: Sensors and Actuators:

Introduction to Sensors: Workflow of a Sensor in a typical system, Classification of Sensors, Sampling DAC and ADC conversion.

Introduction to Actuators: Workflow of an Actuator in a typical system, Classification of Actuators.

Self-Study: Types of Sensors, Interfacing concepts to embedded systems.

Unit III

Communications and networking aspects of IoT:

High bandwidth networking: Ethernet, gigabit Ethernet, Ethernet topologies like bridge and switches, Passive optical fiber network and topologies, WiFi and WiMax. WiFi routers, radius servers, Wireless security with WPA-2, LEAP, enterprise WPA networks

Low Bandwidth Wireless Networks: FSK, LoRa modulation basics, LoRaWAN basics. **Peripherals networking:** Basics of I2C, SPI, RS232, RS485 and CAN bus, Comparisons and use cases of I2C, SPI, RS232, RS485and CAN bus.

Self-Study: Introduction to BLE 5 and industrial Wireless sensor networks, Security in low bandwidth wireless networks, Security in peripheral networks.

Unit IV

Software and middleware aspects of IoT:-

Middleware: Components of Middleware, Types of Databases, Micro services and API's. IP Communication protocols: HTTP, AMQP, MQTT and STOMP.
Self-Study: Protocol definitions, use cases and differences.

Unit V

IoT Platform Design Methodology and Domain Specific IoT.
Self-Study: Futuristic view of IoT, problems pertaining to implementation like scaling, feasibility and management.

Textbook:

1. Srinivasa K G, Siddesh G.M and HanumanthaRaju R "Internet of Things", CENGAGE Leaning India, 2017. (ISBN:978-93-868-5895-5).

References:

1. Vijay Madisetti and Arshdeep Bahga, "Internet of Things (A Hands-on-Approach)", 1stEdition, VPT, 2014. (ISBN: 978-8173719547).
2. Designing the Internet of Things by Adrian McEwenSmart Cities, Software above the level of a single device,Ebooks on IoT byO'Reilly
3. Sentilo middleware http://www.sentilo.io/xwiki/bin/view/Sentilo.About.Product/Whatis
4. Mosquitto broker https://mosquitto.org/
5. Getting started with raspberry pi https://www.raspberrypi.org/resources/learn/
6. Arduino basics https://www.arduino.cc/en/Tutorial/HomePage
7. Wired peripheral protocols http://www.comm.pub.ro/dicm/C7_Serial_Bus.pdf
8. OneM2m http://www.indiaeu-ictstandards.in/wp-content/uploads/2017/04/oneM2M- for-smart-city-TSDSI-presentation-April-21st-2017-Omar-Elloumi.pdf
9. LoRa Modulation http://www.semtech.com/images/datasheet/an1200.22.pdf

Course Outcomes (COs):

At the end of the course, students should be able to:

1. Understand the basics of IoT.
2. Demonstrate various components, layouts and views in creating IoT applications.
3. Design applications using sensors and actuators.
4. Demonstrate the working of long running tasks in the background using IoT.
5. Demonstrate how to write applications for smart world.

Web Technologies

Course Code: CSOE04 **Credits: 3:0:0**
Prerequisites: **Contact Hours:42**
Course Coordinator/s: Dr. Geetha J

Course Contents

Unit I

HTML5 and Java script Basics: Introduction, Cascading Styles Sheet: Concept of CS, Creating Style Sheet, CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts), CSS Id and Class, Box Model (Border, Padding, Margin properties), CSS Advanced (Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class)
JavaScript syntax, Types of Data and Variables, Operations and calculations, The Document Object, Using Events.

Unit II

JavaScript Advanced: Scopes and Closures, Understand "this" and prototypes, OO concepts as applied to JS and prototypal inheritance, Understanding the meaning of asynchronous. Event loops, Promises.

Unit III

Introduction to Server-side JS Framework – Node.js : Introduction - What is Node.js, Architecture, Feature of Node JS, Installation and setup - Creating web servers with HTTP (Request & Response), Understand dependence management: npm and package.json File system APIs

Unit IV

CRUD Operations using Node.js:
Event Handling - GET & POST implementation, Use Express.js to create a REST API. Use GET, POST. Connect to NoSQL MongoDB Database using Node.js, Implementation of CRUD operations.

Unit V

Introduction to Client-side JS Framework – Basics of TypeScript and Angular 4.0
Introduction to TypeScript, Architecture of Angular apps and development setup,

Data and Event handling, Components, Directives, Forms, Consuming HTTP services.

Text Books:

1. Nate Murray, Felipe Coury, Ari Lerner and Carlos Taborda, “ng-book, The Complete Book on Angular 4” September 2016
2. Krasimir Tsonev, “Node.js by Example Paperback”, May 2015

Reference Links:

1. Web link for Angular4.0: https://angular.io/
2. Web link for Node.js : https://nodejs.org/en/
3. Web link for MongoDB: https://www.mongodb.com/

Course Outcomes (COs):

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

1. Describe the functions available for file I/O and changing the properties of the file in Unix OS.
2. Explain the creation of new process, process accounting and process termination.
3. Illustrate the basic IPC issues and techniques in UNIX system programming
4. Explore Node.js and Angular features and create component based web pages using them
5. Design Front-end web pages and connect to the Back-end Databases.

Object Oriented Programming with C++

Course Code: CSOE05 **Credits: 3:0:0**
Prerequisites: **Contact Hours:42**
Course Coordinator/s: Pradeep Kumar D

Course Contents

Unit 1

Beginning with C++ and its features: What is C++?, Applications and structure of C++ program, Different Data types, Variables, Different Operators, expressions, operator overloading and control structures in C++ (Topics from Ch -2,3 of Text). L1, L2

Unit 2

Functions, classes and Objects: Functions, Inline function, function overloading, friend and virtual functions, Specifying a class, C++ program with a class, arrays within a class, memory allocation to objects, array of objects, members, pointers to members and member functions (Selected Topics from Chap-4,5 of Text). L1, L2, L3

Unit 3

Constructors, Destructors and Operator overloading: Constructors, Multiple constructors in a class, Copy constructor, Dynamic constructor, Destructors, Defining operator overloading, Overloading Unary and binary operators, Manipulation of strings using operators (Selected topics from Chap-6, 7 of Text). L1, L2, L3

Unit 4

Inheritance, Pointers, Virtual Functions, Polymorphism: Derived Classes, Single, multilevel, multiple inheritance, Pointers to objects and derived classes, this pointer, Virtual and pure virtual functions (Selected topics from Chap-8,9 of Text). L1, L2, L3

Unit 5

Streams and Working with files: C++ streams and stream classes, formatted and unformatted I/O operations, Output with manipulators, Classes for file stream operations, opening and closing a file, EOF (Selected topics from Chap-10, 11 of Text). L1, L2, L3

Text Book:

1. Object Oriented Programming with C++, E. Balaguruswamy, TMH, 6th Edition, 2013.

Reference Book:

1. Object Oriented Programming using C++, Robert Lafore, Galgotia publication 2010

Course Outcomes (COs):

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

1. Explain the basics of Object Oriented Programming concepts.
2. Apply the object initialization and destroy concept using constructors and destructors.
3. Apply the concept of polymorphism to implement compile time polymorphism in programs by using overloading methods and operators.
4. Use the concept of inheritance to reduce the length of code and evaluate the usefulness. Apply the concept of run time polymorphism by using virtual functions, overriding functions and abstract class in programs.
5. Use I/O operations and file streams in programs.

Introduction to Data structures and algorithms

Course Code: CSOE06 **Credits: 3:0:0**
Prerequisites: C Programming **Contact Hours:42**
Course Coordinator/s: Dr. Sangeetha J

Course Contents

Unit I

Python Overview: The Python Interpreter, Preview of a Python Program, Identifiers, Objects, and the Assignment Statement, Creating and Using Objects, Python's Built-In Classes, Compound Expressions and Operator Precedence, Conditionals, Loops, Functions, Python's Built-In Functions, Simple Input and Output, Console Input and Output, Iterators and Generators, Conditional Expressions. (Text Book 1- Chapter 1, 1.1 to 1.6.1)

Unit II

Arrays, Stacks, Queues: Stacks: Low level arrays, Python's List and Tuple Classes, Python's String Class. The Stack Abstract Data Type, Stack implementation, **Queues:** The Queue Abstract Data Type, Queue implementation. (Text Book 1- Chapter 5, 5.2, 5.3.3, 5.4, Chapter 6, 6.1.1, 6.1.2, 6.2)

Unit III

Linked Lists: Singly Linked Lists: Implementing a Stack with a Singly Linked List Implementing a Queue with a Singly Linked List, Circularly Linked Lists: Implementing a Queue with a Circularly Linked List (Text Book 1-Chapter 7, 7.1, 7.2)

Unit- IV

Searching Techniques, Sorting Techniques, and Trees: Searching Techniques: Linear search, Binary search and their implementation. **Sorting algorithms:** Bubble sort, Selection sort, **Trees:** Binary Trees, Tree Structure, The Binary Tree, Properties, and Implementation of Tree Traversals. (Text Book 2-Chapter 5, 5.1, 5.2, Chapter 13, 13.1, 13.2)

Unit V

Text Processing, Graph Algorithms: Text Processing: Brute-Force Methods **Graph Algorithms:** Graphs, The Graph ADT, Data Structures for Graphs,

Python Implementation, Graph Traversals, Depth-First Search, Breadth-First Search, Dijkstra's Algorithm (Text Book 1- Chapter 13, 13.2.1, Chapter 14, 14.1 to 14.3, 14.6.2)

Text Books:

1. Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, Wiley, 2013.
2. Data structures and Algorithms using Python, Rance D Necaise, Wiley Publishing, 2010.

Reference Books:

1. How to think like a computer scientist: learning with python. Downey, A., Elkner, J., & Meyers, C., Green Tea Press, Wellesley, Massachusetts, 2002.
2. Problem Solving with Algorithms and Data Structures Release 3.0, Brad Miller, David Ranum, 2013
3. Data Structure and Algorithmic Thinking with Python. Karumanchi, Narasimha, CareerMonk Publications, 2016

Course Outcomes (CO's):

At the end of the course, student should be able to:

1. Understanding basics of Python and its built in functions.
2. Implement storage and retrieval using arrays, stacks and queues.
3. Implement storage and retrieval in linked list and to implement stacks and queues using linked list.
4. Learn to implement searching, sorting techniques and to implement storage and retrieval of binary trees.
5. Illustration of text processing techniques for pattern matching and also to develop solutions for problems based on graph algorithms

Programming in Java

Course Code: CSOE07 **Credits: 3:0:0**
Prerequisites: OOPS concepts **Contact Hours:42**
Course Coordinator/s: Hanumantharaju R

Course Contents

Unit I

Introduction to Java Programming: Java Buzzwords, Overview of Java Datatypes, Variables, arrays, Control statements. Java Programming Fundamentals: Object-Oriented Programming, the Three OOP Principles, Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods

Unit II

Java Programming Fundamentals: Constructors, The this Keyword, Garbage Collection, The finalize( ) Method, Overloading Methods, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Introducing Access Control, Understanding static, Introducing final, Introducing Nested and Inner Classes

Unit III

Inheritance, Packages & Interfaces: Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Called, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, Packages, Access Protection, Importing Packages, Interfaces, String and String Buffer Handling.

Unit IV

Exception Handling: Exception-Handling Fundamentals, Exception Classes, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch clauses, Nested try Statements, throw, throws, finally. Multithreaded Programming: Java Thread Classes, The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities, Synchronization, Suspending, Resuming and Stopping Threads.

Unit V

Event Handling, Introducing Swing: 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. Swing: Introducing Swing. Lambda Expressions: Fundamentals, Block Lambda expressions, Passing Lambda Expressions as Argument, Lambda Expressions and Exceptions, Method References.

Text Book:

1. The Complete Reference - Java, Herbert Schildt 10th Edition, 2017, TMH Publications, ISBN: 9789387432291.

Reference Book:

1. Head First Java, Kathy Sierra and Bert Bates, 2nd Edition, 2014, Oreilly Publication, ISBN: 9788173666.

Operating Systems

Course Code: CSOE08 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours: 42**
Course Coordinator/s: Chandrika Prasad

Course Contents

Unit 1

Introduction to Operating Systems- What operating systems do, Operating System operations, Process management, Memory management, Storage management, Protection and security. **System Structures**- Operating System Services, System calls, Operating System design and implementation, Operating System structure, Virtual machines,

Unit 2

Process Management - Process concept, Process scheduling, Operations on processes, Inter- process communication. **Process Scheduling**- Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple-Processor scheduling. **Process Synchronization-** Background, The Critical section problem, Synchronization hardware, Semaphores, Classical problems of synchronization.

Unit 3

Deadlocks- System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection and recovery from deadlock. **Memory Management Strategies**- Background, Swapping, Contiguous memory allocation, Paging, Structure of page table, Segmentation. **Virtual Memory Management-** Background, Demand paging, Page replacement, Thrashing.

Unit 4

File System- File concept, Access methods, Directory structure, Protection. **Implementing File System**- File system structure, File system implementation, Directory implementation, Allocation methods, Free space management. **Secondary Storage Structures**- Mass storage structures, Disk structure, Disk scheduling.

Unit 5

Dockers- Docker Basics and Architecture, what is containerization, How are containers different from physical machines and VMs, Docker evolution and architecture, Developments in Docker world, Docker tooling, Basic Docker commands. **Docker Networking**- Introduction, Types of Docker networks, Using Networks, Identifying container networks, **Docker Volumes**- Managing data in Docker containers with volumes, Volume file systems and basic Docker image file systems, Creating and managing volumes.

Textbook:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating System Principles, 8th edition, Wiley-India, 2009.

Reference Books:

1. D.M Dhamdhere: Operating systems - A concept-based Approach, 2nd Edition, Tata McGraw- Hill, 2002.
2. P.C.P. Bhatt: Operating Systems, 2nd Edition, PHI, 2006.
3. Harvey M Deital: Operating systems, 3rd Edition, Addison Wesley, 1990.

Course Outcomes (COs):

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

1. Describe operating system operations and operating system structures.
2. Assess different scheduling algorithms and concurrency control techniques to provide co-ordination among threads and processes.
3. Examine various methods for handling deadlocks and memory management techniques.
4. Identify file systems and recognize its implementation and secondary storage structure
5. Illustrate Dockers techniques for various networking and file system applications.

Introduction to Artificial Intelligence and Machine Learning

Course Code: CSOE09 **Credits: 3:0:0**

Contact Hours:42

Prerequisites: Knowledge of any advanced programming language, Algorithms and Data structures, Elementary Discrete Mathematics or similar.

Course Coordinator/s: Dr. Annapurna P Patil and Shreekant Jere

Course Contents

Unit I

Introduction: What is AI? Foundation and History of Artificial Intelligence. **Intelligent Agents:** Agents and Environments, Rationality, The Nature of Environments, The Structure of Agents. **Problem-solving by search:** Problem-Solving Agents, Example Problems, Searching for Solution, Uniformed Search Strategies, Informed Search Strategies, Heuristic Functions. (Chapter 1, 2, 3 of Textbook 1)

Unit II

Logical Agents: Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic, Reasoning patterns in propositional Logic, Effective Propositional Model Checking, Agents Based on Propositional Logic. **First-Order Logic**: Representation Revisited, Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic. **Interference in First-order Logic**: Propositional vs. First-Order Inference, Unification and Lifting, Forward chaining, Backward chaining, Resolution. (Chapter 7, 8, 9 of Textbook 1)

Unit III

Planning: Definition, Planning with State-Space Search, Planning Graphs, Other Planning Approaches Analysis. **Uncertainty:** Acting under Uncertainty, Basic Probability Notations, Inference using Full Joint Distributions, Independence, Bayes' Rule and its Use. **Learning from Examples**: Forms of Learning, Supervised Learning, Learning Decision Trees, Artificial Neural Networks, Support Vector Machines, Ensemble Learning. (Chapter 10, Chapter 13, Chapter 18.1,18.2,18.3, 18.7,18.9,18.10 of Textbook 1)

Unit IV

Natural Language Processing: Language Models, Text Classification, Information Retrieval, Information Extraction. **Natural Language communication:** Phrase Structure Grammars, Syntactic Analysis, Augmented Grammars and Semantic Interpretation, Machine translation, Speech recognition. (Chapter 22, 23 of Textbook 1)

Unit V

Genetic Algorithms: Genetic Algorithms Introduction, Significance of Genetic Operators, Termination Parameters, Niching and Speciation, Evolving Neural Networks, Theoretical Grounding, Ant Algorithms. **Robotics:** Introduction, Hardware, Perception, Planning to Move, Planning Uncertain Movement, Moving, Robotic Software Architecture, Application Domains. **Philosophical Foundations:** Weak and Strong AI, The Ethics and Risks of Developing AI, **AI: The present and Future**. (Chapter 23 of Textbook 2, Chapter 25, 26 ,27 of Text Book 1)

Textbooks:

1. Stuart Russel, Peter Norvig: Artificial Intelligence - A Modern Approach, 3rd Edition, Pearson Education, 2012. (Unit-1,2,3,4,5).
2. Elaine Rich, Kevin Knight, Shivashankar B Nair: Artificial Intelligence, 3rd Edition, Tata McGraw Hill, 2011. (Unit-5).

Reference Books:

1. Peter Jackson, "Introduction to Expert Systems", 3rd Edition, Pearson Education, 2007. (Unit 5).
2. Deepak Khemani "Artificial Intelligence", Tata Mc Graw Hill Education 2013. (unit 3).
3. http://nptel.ac.in

Course Outcomes (COs):

At the end of the course, the student should be able to:

1. Identify problems that are amenable to specific solution by appropriate AI methods.
2. Utilize various symbolic knowledge representation to specify domains and reasoning tasks of a situated software agent. Use different logical

systems for inference over formal domain representations, and trace how an inference algorithm works on a given problem specification.

3. Formalize a given problem in the language/framework of different AI methods and solve using basic AI algorithms.
4. Design and carry out an empirical evaluation of different algorithms on a problem formalization and state the conclusions that the evaluation supports.
5. Communicate scientific knowledge at different levels of abstraction in a variety of research settings.

Introduction to Big Data Analytics

Course Code: CSOE10 **Credits: 3:0:0**
Prerequisites: **Contact Hours: 42**
Course Coordinator/s: Dr. D S Jayalakshmi

Course Contents

Unit I

UNDERSTANDING BIG DATA: What is big data – why big data –.Data!, Data Storage and Analysis, Comparison with Other Systems, Rational Database Management System , Grid Computing, Volunteer Computing, convergence of key trends – unstructured data – industry examples of big data – web analytics – big data and marketing – fraud and big data – risk and big data – credit risk management – big data and algorithmic trading – big data and healthcare – big data in medicine – advertising and big data – big data technologies – introduction to Hadoop – open source technologies – cloud and big data – mobile business intelligence – Crowd sourcing analytics – inter and trans firewall analytics.

Unit II

NOSQL DATA MANAGEMENT: Introduction to NoSQL – aggregate data models – aggregates – key-value and document data models – relationships – graph databases – schema less databases – materialized views – distribution models – shading — version – map reduce – partitioning and combining – composing map-reduce calculations.

Unit III

BASICS OF HADOOP: Data format – analyzing data with Hadoop – scaling out – Hadoop streaming – Hadoop pipes – design of Hadoop distributed file system (HDFS) – HDFS concepts – Java interface – data flow – Hadoop I/O – data integrity – compression – serialization – Avro – file-based data structures.

Unit IV

MAPREDUCE APPLICATIONS: MapReduce workflows – unit tests with MRUnit – test data and local tests– anatomy of MapReduce job run – classic Map-reduce – YARN – failures in classic Map-reduce and YARN – job scheduling – shuffle and sort – task execution – MapReduce types – input formats – output formats

Unit V

HADOOP RELATED TOOLS: Hbase – data model and implementations – Hbase clients – Hbase examples – praxis. Cassandra – Cassandra data model – Cassandra examples – Cassandra clients –Hadoop integration. Pig – Grunt – pig data model – Pig Latin – developing and testing Pig Latin scripts. Hive – data types and file formats – HiveQL data definition – HiveQL data manipulation – HiveQL queries.

Textbooks:

1. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012.
2. Eric Sammer, "Hadoop Operations", O'Reilley, 2012.

Reference Books:

1. Vignesh Prajapati, Big data analytics with R and Hadoop, SPD 2013.
2. E. Capriolo, D. Wampler, and J. Rutherglen, "Programming Hive", O'Reilley, 2012.
3. Lars George, "HBase: The Definitive Guide", O'Reilley, 2011.
4. Alan Gates, "Programming Pig", O'Reilley, 2011

Course Outcomes (COs):

At the end of the course, the student should be able to:

1. Identify the role of Big Data.
2. Utilize unstructured database in managing the data.
3. Formalize basics of Hadoop.
4. Understand the various Mapreduce applications.
5. Interpret on various Hadoop related tools.