RAMAIAH
Institute of Technology

CURRICULUM

Academic year 2022 – 2023

COMPUTER SCIENCE AND ENGINEERING

VII & VIII 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 also been conferred autonomous status for Ph.D program since 2021. 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 65% 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 & Centre for Bio and Energy Materials Innovation. **M S Ramaiah Institute of Technology has obtained "Scimago Institutions Rankings" All India Rank 107 & world ranking 600 for the year 2022.**

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 is recognized by Atal Ranking of Institutions on Innovation Achievements (ARIIA), 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 67th rank among 1249 top Engineering Institutions & 17th Rank for School of Architecture in India for the year 2022 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. 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 SAP, IBM and HPE. The department conducts subjects with more of hands- on sessions and encourages students to take up MOOC based online courses in NPTEL, IIT Bombay, 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

- Imparting quality technical education by nurturing a conducive learning environment through continuous improvement and customization.
- Establishing research clusters in emerging areas in collaboration with globally reputed organizations.
- 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 graduate studies.

PEO3 Be able to work effectively in multidisciplinary environments and be responsible members / 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

VII SEMESTER

SI. No	Course Code	Course Name	Page No.	Category	Credits				Contact Hours
					L	T	P	Total	
1.	CS71	Cloud Computing and Big data	1-3	PCC	4	0	0	4	56
2.	CS72	Multicore Architecture and Programming	4-5	PCC	3	1	0	4	42+28
3.	CS73	Economics and Management	6-7	HSMC	3	0	0	3	42
4.	CSE74x	Professional Elective-4		PEC	*	*	*	3	--
5.	CSE75x	Professional Elective-5		PEC	*	*	*	3	--
6.	CSL76	Big Data Laboratory	30	PCC	0	0	1	1	28
7.	CSL77	Micro Services laboratory	31	PCC	0	0	1	1	28
8.	CSSE	Seminar	32	PCC	0	0	1	1	28
Total					-	-	-	20	

List of Professional Elective for VII Semester

	Professional Elective-4		Page No.	Professional Elective-5		Page No.
SI. No	Course Code	Course Name		Course Code	Course Name	
1.	CSE742	Storage Area Networks	8-9	CSE751	Distributed Systems	19-21
2.	CSE743	Introduction to DevOps	10-12	CSE752	Cyber Physical Systems	22-23
3.	CSE744	Network Management	13-14	CSE753	Software Testing	24-25
4.	CSE745	Multimedia Computing	15-16	CSE754	Introduction to Deep Learning	26-27
5.	CSE746	Blockchain Essentials & DApps	17-18	CSE756	Soft Computing	28-29

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

VIII SEMESTER

SI. No	Course Code	Course Name	Page No.	Category	Credits			
					L	T	P	Total
1	CSIN	Internship	33-34	IN	*	*	*	3
2	CSP	Project Work	35-39	PW	0	0	14	14
Total					--	--	--	**17**

Cloud Computing and Big data

Course Code: CS71 **Credits: 4:0:0**
Prerequisites: Nil **Contact Hours: 56**
Course Coordinator/s: Dr. S Rajarajeswari

Course Contents

Unit I

Introduction: Cloud Computing, delivery models & Services, Ethical issues, Cloud vulnerabilities, Challenges, Cloud Infrastructure: Amazon, Google, Azure & online services, open source private clouds. Storage diversity and vendor lock-in, intercloud, Energy use & ecological impact of data centers, Service and Compliance level agreement, Responsibility sharing, User experience, Software licensing. Applications & Paradigms: Challenges, existing and new application opportunities, Architectural styles of cloud applications; different cloud architectures, Applications: Healthcare, Energy systems, transportation, manufacturing, Education, Government, mobile communication, application development.

Unit II

Cloud Resource Virtualization: Layering and virtualization, Virtual machine monitors, Virtual machines, VM Performance and security isolation, virtualization types, Hardware support for virtualization, A performance comparison of virtual machines, The darker side of virtualization, Software fault isolation.

Cloud Resource Management and Scheduling: Policies and mechanisms for resource management, Resource bundling, combinatorial auctions for cloud Scheduling algorithms for computing clouds, fair queuing, start time fair queuing, borrowed virtual time Resource management and application scaling.

Unit III

Cloud Security: Risks, privacy and privacy impacts assessments, Trust, OS, VM security, security of virtualization, risk posed by shared images, mgmt OS, Xoar, and Trusted VMM.

Introduction to Big data: Types of digital data; Big data – definition, characteristics, evolution of Big data, Challenges; Comparison with BI ; Cloud Computing and Big Data, Cloud Services for Big Data, In-Memory Computing Technology for Big Data.

Unit IV

Workflows: coordination of multiple activities, Coordination based on a state machine model -the Zoo Keeper, Apache Hadoop – Introduction, Architecture and Components, HDFS, The Map Reduce programming model, YARN, Interacting with Hadoop ecosystem – Pig, Hive, HBase, Sqoop, A case study: the GrepTheWeb application.

Unit V

Analysing Big Data: The Challenges of Data Science, Introducing Apache Spark. Introduction to Data Analysis with Scala and Spark: Scala for Data Scientists, The Spark Programming Model, Record Linkage, Getting Started: The Spark Shell and Spark Context, Bringing Data from the Cluster to the Client, Shipping Code from the Client to the Cluster, Structuring Data with Tuples and Case Classes, Aggregations, Creating Histograms, Summary Statistics for Continuous Variables, Creating Reusable Code for Computing Summary Statistics, Simple Variable Selection and Scoring.

Text Books:

1. Cloud Computing Theory and Practice – DAN C. Marinescu – Morgan Kaufmann Elsevier.
2. Big Data Analytics, Seema Acharya and Subhashini Chellappan. 2nd edition, Wiley India Pvt. Ltd. 2019.
3. Sandy Ryza, Uri Laserson, Josh Wills, Sean Owen Advanced Analytics with Spark 2nd Edition, Publisher: O'Reilly Media.

Reference Books/Links:

1. Cloud Computing: Theory and Practice, Dan Marinescu, 1st edition, MK Publishers, 2013.
2. Distributed and Cloud Computing, From Parallel Processing to the Internet of Things, Kai Hwang, Jack Dongarra, Geoffrey Fox. MK Publishers.
3. Cloud Computing: A Practical Approach, Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, McGraw Fill, 2010.
4. Cloud Computing A hands - on approach – Arshdeep Bahga & Vijay madisetti Universities press.
5. Internet – White papers from IBM, other enterprise resources.
6. Mining of Massive Datasets. 2nd edition. - Jure Leskovec, Anand Rajaraman, Jeff Ullman. Cambridge University Press. http://www.mmds.org/
7. https://docs.rightscale.com/cm/designers_guide/cm-cloud Computing-system-architecture-diagrams.html.

Course Outcomes (COs):

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

1. Analyze the Delivery model, services, and applications of Cloud Computing. (PO-1, 2, 3, 4,5,6,9,10,11,12, PSO-3)
2. Analyze different cloud resource virtualization types, resource management, and Scheduling. (PO-1, 2,3,4,5,8,9,10,11, PSO-3)
3. Demonstrate Big Data concepts and identify the risks with respect to cloud security. (PO-1, 2,3,4,5,6,7,8,9,12, PSO-3)
4. Create different Workflow for coordination services and Analyze different databases used for Hadoop/Map-reduce framework. (PO-1,2,3,4,56,7,8,9,10,11,12, PSO-3)
5. Analyze the Big data using the Spark programming model. (PO-1,2,3,4,5,7,8,9,10,11,12, PSO-3)

Multicore Architecture and Programming

Course Code: CS72 **Credits: 3:1:0**

Contact Hours: 42+28

Prerequisites: Computer Organization and Unix System Programming

Course Coordinator/s: Mallegowda M

Course Contents

UNIT I

Introduction to High–Performance Computers, Memory Hierarchy, and CPU Design: Reduced Instruction Set Computers, Multiple–Core Processors, Vector Processors, Parallel Semantics, Distributed Memory Programming.

UNIT II

Programming Shared Address Space Platforms: Thread Basics, Why Threads? The POSIX Thread API, Thread Creation and Termination, Synchronization Primitives in Pthreads, Controlling Thread and Synchronization Attributes, Thread Cancellation, Composite Synchronization Constructs, for Designing Asynchronous Programs, OpenMP: a Standard for Directive Based Parallel Programming.

UNIT III

Programming using the Message-Passing Paradigm: Principles of Message- Passing Programming, The Building Blocks: Send and Receive Operations, MPI: the Message Passing Interface, Topologies and Embedding, Overlapping Communication with Computation, Collective Communication and Computation Operations**,** Groups and Communicators.

UNIT IV

Introduction: GPUs as Parallel Computers, Architecture of a Model GPU, Why More Speed or Parallelism? Parallel Programming Languages and Models, Overarching Goals. History of GPU Computing: Evolution of Graphics Pipelines, GPU Computing. Introduction to CUDA: Data Parallelism, CUDA Program Structure, a Matrix-Matrix Multiplication Example, Device Memories and Data Transfer, Kernel Functions and Threading.

Unit V

CUDA Threads: CUDA Thread Organization, Using blockIdx and threadIdx, Synchronization and Transparent Scalability, Thread Assignment, Thread Scheduling and Latency Tolerance. CUDA Memories: Importance of Memory Access Efficiency, CUDA Device Memory Types, A Strategy for Reducing Global Memory Traffic, Memory as a limiting Factor to Parallelism. Performance Considerations: More on Thread Execution, Global Memory Bandwidth, Dynamic Partitioning of SM Resources, Data Perfecting, Instruction Mix, Thread Granularity, Measured Performance and Summary.

Text Books:

1. Rubin H Landau, Oregon State University, http://science.oregonstate.edu/ rubin/.
2. Ananth Grama, Anshul Gupta, Vipin kumar, George Karypis Introduction to parallel computing, second edition, 2015, Pearson education publishers.
3. David B Kirk, Wen-mei W. Hwu, "Programming Massively Parallel Processors – A Hands-on Approach", First Edition, Elsevier and nvidia Publishers, 2015.

Reference Books:

1. Thomas Rauber and Gudula Runger Parallel Programming for Multicore and cluster systems,Springer International Edition, 2009 .
2. Hennessey and Patterson Computer Architecture: A quantitative Approach, Morgan Kaufman Publishers
3. Michael J.Quin "Parallel Programming in C with MPI and Open MP", McGraw Hill.
4. Peter S. Pacheco, - An Introduction to Parallel Programming, Morgan- Kauffman / Elsevier, 2011.

Course Outcomes (COs):

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

1. Explain the technologies and architectures for parallel computing.(PO-1, 8,9,10,11, PSO-2)
2. Design and develop parallel programs using OpenMp programming interface. (PO-1,2,3,4,5,8,9,10, PSO-2)
3. Discuss the principles and architecture of message-passing programming. (PO-1,2,3,4,5,8,9,10, PSO-2)
4. Describe Graphical Processing Units and architecture. (PO-1,2,3,4,5,8,9,10, PSO-2)
5. Analyze the features GPUs, their functionalities and also Design parallel applications using CUDA-C. (PO-1,2,3,4,5,8,9,10, PSO-2)

Economics and Management

Course Code: CS73 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours: 42**
Course Coordinator/s: Veena G S

Course Contents

Unit I

Introduction to Engineering Economics: Engineering Decision Makers, Engineering and Economics, Economics: A Capsule View, Problem Solving and Decision Making. Time Value of Money: Interest and the Time Value of Money, Reasons for Interest, Simple Interest, Compound Interest, Time Value Equivalence, Compound Interest Factors, Cash Flow Diagrams, Calculation of Time Value Equivalences. Present Worth Comparisons: Conditions for Present Worth Comparisons, Basic Present worth Comparison Patterns, Comparison of Assets that have unequal lives, Comparison of Assets assumed to have infinite lives.

Unit II

Present Worth Comparisons: Comparison of deferred investments, Future worth comparisons, Valuation, Payback Comparison Method. Equivalent Annual Worth Comparisons: Utilization of Equivalent Annual Worth Comparisons, Consideration of Asset Life, Use of a sinking fund, Equivalent uniform payments when interest rates vary, Annuity contract for a guaranteed income.

Unit III

Rate of Return Calculations: Rate of Return, Minimum Acceptable rate of return, internal rate of return, Consistency of IRR with other economic comparison methods, IRR Misconceptions, Final comments on theory and practice behind interest rates.
Introduction to Project Management: What is project and project management? Role of project manager, a system view of project management, project phases and project cycle, Context of IT projects, Strategic Planning and Project Selection: Preliminary scope statements, project management plans, project execution, monitoring and control of project work.

Unit IV

Project Management Techniques: Identifying organizational structures, estimating costs and budgeting, using critical path project management tools, Establishing the critical path, Tracking project milestones, Using the programme evaluation and review technique, Using process improvement tools.

Unit V

Project quality management: Importance of quality management, what is quality management, planning, assurance, control, tools and techniques for quality control. Project communication management: Importance, communication planning, information distribution. Project risk management: what is risk management, risk management planning, common source of risk in IT, risk identification, risk monitoring and control.

Text Books:

1. James L Riggs, David D Bedworth, Sabah U Randhawa: Engineering Economics, Fourth Edition, TMH, 1996.
2. Kathy Schwalbe: Project Management in IT, India edition, Cengage Learning, 2007.

Reference Books:

1. R. Panneerselvam: Engineering Economics, PHI Learning Pvt. Ltd., 2001.
2. Bob Hughes, Mike Cotterell: Software Project Management, Tata McGraw Hill, 2006.
3. Pankaj Jalote: Software Project Management in Practice, Pearson,2006.

Course Outcomes (CO's):

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

1. Describe the basic concepts of engineering economics and Time Value Equivalence of money. (PO-1,2,3,4,11,12, PSO-2)
2. Calculate present worth, future worth and equivalent annual worth of investments and compare investment alternatives. (PO-1,5,7,9,12, PSO-2)
3. Identify the various rates of returns (PO-3,4,5,7,9,10,12, PSO-2)
4. Estimate the time, scope and cost of a software project. (PO-1,3,5, 6, 8, 9, 10, 11, 12, PSO-3)
5. Identify various quality issues, communication issues and risks in a software project. (PO-3,4,6,7,9,10,11,12, PSO-3)

Storage Area Networks

Course Code: CSE742 **Credits: 3:0:0**

Contact Hours: 42

Prerequisites: Computer Networks, Computer Organization, Operating Systems

Course Coordinator/s: Dr. Divakar Harekal

Course Contents

Unit I

Introduction: Information Storage, Evolution of Storage Architecture, Data Centre Infrastructure, Virtualization and Cloud Computing. Data Centre Environment: Application, DBMS, Host, Connectivity, Storage, Disk Drive Components, Disk Drive Performance, Host Access to Data, Direct-Attached Storage, Storage Design Based on Application, Disk Native, Command Queuing, Introduction to Flash Drives.

Unit II

Data Protection: RAID Implementation Methods, Array Components, Techniques, Levels, Impact on Disk Performance, Comparison, Hot Spares. Intelligent Storage System: Components, Storage Provisioning, Types.

Unit III

Fiber Channel Storage Area Networks: FC Overview, Evolution, Components, FC Connectivity, Ports, FC Architecture, Fabric Services, Login Types, Zoning, FC Topologies, Virtualization in SAN.IP SAN and FCoE: iSCSI,FCIP, FCoE.

Unit IV

Network-Attached Storage: Benefits, Components, NAS I/O Operation, Implementations, File Sharing Protocols, I/O Operations, Factors Affecting NAS Performance, File-Level Virtualization Object Based and Unified Storage: Object Based Storage Devices, Content Addressed Storage, CAS Use Cases, Unified Storage, Backup Archive and Replication.

Unit V

Business Continuity: Information Availability, Terminology, Planning Lifecycle, Failure Analysis, Impact Analysis, Challenges, Adoption Considerations. Monitoring, Management Activities, Management Challenges, Information Lifecycle Management, Storage Tiering, Domains Managing the Storage Infrastructure: Securing the Storage, Infrastructure: Framework, Risk Triad.

Text Book:

1. EMC Education Services, edited by Somasundaram G., Alok Shrivastava "Information Storage and Management", 2nd edition, Wiley India, 2012.

Reference Books:

1. Ulf Troppens, Rainer Erkens and Wolfgang Muller: Storage Networks Explained, 1st Edition, Wiley India, 2012.
2. Robert Spalding: Storage Networks, The Complete Reference, 1st Edition, Tata McGraw Hill, 2011.

Course Outcomes (COs):

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

1. Identify the need for storage centric network and its benefits of its adoption (PO- 1, 2, 3, 4, PSO-1,2)
2. Design a storage solution for an application depending on the IOPS and RAID requirements (PO- 1, 2, 3, 4, PSO-1,2)
3. Describe the Fiber channel stack and working of the different layers (PO- 1, 2, 3, 4, PSO-1,2)
4. Summarize about NAS, object oriented storage and backup and recovery (PO- 1, 2, 3, 4, PSO-1,2)
5. Develop a business continuity plan and ILM of an enterprise (PO- 1, 2, 3, 4, PSO-1,2)

Introduction to DevOps

Course Code: CSE743 **Credits: 3:0:0**

Contact Hours: 42

Prerequisites: Pragramming Language, Basics of Cloud Computing

Course Coordinator/s: Dr. Geetha J & Pramod Sunagar

Course Contents

Unit I

DevOps and Infrastructure as Code: DevOps Culture and Practices: Getting started with DevOps, Implementing CI/CD and continuous deployment, Understanding IaC practices. Using Ansible for Configuring IaaS Infrastructure: Installing Ansible, Creating an inventory for targeting Ansible hosts, writing the first playbook, Executing Ansible, Protecting data with Ansible Vault, Using a dynamic inventory for Azure infrastructure. Optimizing Infrastructure Deployment with Packer: An overview of Packer, Creating Packer templates for Azure VMs with scripts, Using Ansible in a Packer template, Executing Packer, Using a Packer image with Terraform.

Unit II

DevOps CI/CD Pipeline: Managing Your Source Code with Git: Overviewing Git and its command lines, Understanding the Git process and GitFlow pattern. Continuous Integration and Continuous Delivery: The CI/CD principles, using a package manager, Using Jenkins, Using Azure Pipelines, Using GitLab CI.

Unit III

Containerized Applications with Docker and Kubernetes: Containerizing Your Application with Docker: Installing Docker, Creating a Dockerfile, Building and running a container on a local machine, Pushing an image to Docker Hub, Deploying a container to ACI with a CI/CD pipeline. Managing Containers Effectively with Kubernetes: Installing Kubernetes, First example of Kubernetes application deployment, Using HELM as a package manager, Using AKS, Creating a CI/CD pipeline for Kubernetes with Azure Pipelines.

Unit IV

Testing Your Application: Testing APIs with Postman: Creating a Postman collection with requests, Using environments and variables to dynamize requests, Writing Postman tests, Executing Postman request tests locally, Understanding the Newman concept, Preparing Postman collections for Newman, Running the Newman command line,

Integration of Newman in the CI/CD pipeline process. Static Code Analysis with SonarQube: Exploring SonarQube, Installing SonarQube, Real-time analysis with SonarLint, Executing SonarQube in continuous integration. Security and Performance Tests: Applying web security and penetration testing with ZAP, Running performance tests with Postman.

Unit V

Taking DevOps Further: Security in the DevOps Process with DevSecOps: Testing Azure infrastructure compliance with Chef InSpec, Using the Secure DevOps Kit for Azure, Preserving data with HashiCorp's Vault. Reducing Deployment Downtime: Reducing deployment downtime with Terraform, Understanding blue-green deployment concepts and patterns, Applying blue-green deployments on Azure. DevOps for Open Source Projects: Storing the source code in GitHub, Contributing using pull requests, Managing the changelog and release notes, Sharing binaries in GitHub releases, Using Travis CI for continuous integration, Getting started with GitHub Actions, Analyzing code with SonarCloud, Detecting security vulnerabilities with WhiteSource Bolt. DevOps Best Practices.

Text Book:

1. Learning DevOps: Mikael Krief, October 2019, Packt Publishing Ltd, ISBN: 978-1-83864-273-0.

Reference Books:

1. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Jez Humble and David Farley, 2010, Addison-Wesley Professional, ISBN: 9780321670250.
2. The DevOPS Handbook: How to Create World, Gene Kim & Jez Humble, 2016, It Revolution Press, ISBN: 9781942788003
3. The Phoenix Project: A Novel about It, Devops, and Helping Your Business Win, George Spafford & Gene Kim, 2018, It Revolution Press, ISBN: 9781942788294
4. Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale, Jennifer Davi & Ryn Daniels, 2016, O'Reilly Media, ISBN

Course Outcomes (COs):

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

1. Demonstrate the DevOps culture by illustrating applications cloud infrastructure and configuration management with Ansible. (PO-2, 3, 5, 9, PSO-3)
2. Apply the DevOps pipeline process starting with continuous integration and continuous deployment principles. (PO-2, 3, 5, 9, PSO-3)
3. Demonstrate how to create and run a container from a Docker file and deploy a complex application on Kubernetes. (PO-2, 3, 5, 9, PSO-3)
4. Illustrate the different ways to test APIs with Postman, static code analysis with SonarQube and perform security and performance tests. (PO-2, 3, 5, 9, PSO-3)
5. Apply security in the DevOps process with DevSecOps and related best practices. (PO-2, 3, 5, 9, PSO-3)

Network Management

Course Code: CSE744 **Credits: 3:0:0**
Prerequisites: Computer Networks **Contact Hours: 42**
Course Coordinator/s: Dr. Shilpa Chaudhari

Course Contents

Unit I

Data communications and Network Management Overview: Analogy of Telephone Network Management, Communications protocols and Standards, Case Histories of Networking and Management, Challenges of Information *T*echnology Managers, Network Management: Goals, Organization, and Functions, Network and System Management, Network Management System Platform, Current Status and future of Network Management.

Unit II

SNMPV1 Network Management: Organization and Information Models, Managed network: Case Histories and Examples, the History of SNMP Management, The SNMP Model, The Organization Model, System Overview, The Information Model. SNMPv1 Network Management: Communication and Functional Models. The SNMP Communication Model, Functional Model.

Unit III

SNMP Management: SNMPv2: Major Changes in SNMPv2, SNMPv2 System Architecture, SNMPv2 Structure of Management Information, SNMPv2 Management Information Base, SNMPv2 Protocol, Compatibility with SNMPv1. SNMP Management: SNMPv3, Introduction, SNMPv3 Key Features, SNMPv3 Documentation Architecture, SNMPv3 Applications, SNMPv3 Management Information Base, Security, SNMPv3 User-based Security Model, Authentication Protocols, Encryption Protocol**.**

Unit IV

SNMP Management: RMON: What is Remote Monitoring?, RMON SMI and MIB, RMON1, RMON2, ATM Remote Monitoring, a Case Study of Internet Traffic Using RMON, ATM Networks.

Unit V

Network Management Tools and Systems: Network Management Tools, Network Statistics Measurement Systems, History of Enterprise Management, Network Management systems, Commercial Network Management Systems, System Management, and Enterprise Management Solutions.

Text Book:

1. Network Management: Principles and Practice, Mani Subramanian, Georgia Institute of Technology, 2000, Pearson.

Course Outcomes (COs):

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

1. Identify five major functional areas of network management. (PO-1,3,4, PSO- 1, 2)
2. Explore SNMP Management Information, Standard MIBs and SNMP protocol. (PO- 1,3,4, PSO- 1, 2)
3. Defend the need for security in networking monitoring and control. (PO-1,3,4, PSO- 1, 2)
4. Devise Remote network monitoring System for statistics collection, alarms and filters. (PO-1,3,4, PSO- 1, 2)
5. Summarize complete network management plan for a moderate to large network enterprise through case studies. (PO-1,3,4, PSO- 1, 2)

Multimedia Computing

Course Code: CSE745 **Credits: 3:0:0**
Prerequisites: Nil **Contact Hours: 42**
Course Coordinator/s: Dr. Shilpa Chaudhari

Course Contents

Unit I

Introduction, Media and Data Streams, Audio Technology: Multimedia Elements, Multimedia Applications, Multimedia Systems Architecture, Evolving Technologies for Multimedia Systems, Defining Objects for Multimedia Systems, Multimedia Data Interface Standards, The need for Data Compression, Multimedia Databases. Media: Perception Media, Representation Media, Presentation Media, Storage Media, Characterizing Continuous Media Data Streams. Sound: Frequency, Amplitude, Sound Perception and Psychoacoustics, Audio Representation on Computers, Three Dimensional Sound Projection, Music and MIDI Standards, Speech Signals, Speech Output, Speech Input, Speech Transmission. Graphics and Images, Video Technology, Computer-Based Animation: Capturing Graphics and Images Computer Assisted Graphics and Image Processing, Reconstructing Images, Graphics and Image Output Options. Basics, Television Systems, Digitalization of Video Signals, Digital Television, Basic Concepts, Specification of Animations, Methods of Controlling Animation, Display of Animation, Transmission of Animation, Virtual Reality Modeling Language.

Unit II

Data Compression: Storage Space, Coding Requirements, Source, Entropy, and Hybrid Coding, Basic Compression Techniques, JPEG: Image Preparation, Lossy Sequential DCT-based Mode, Expanded Lossy DCT-based Mode, Lossless Mode, Hierarchical Mode. H.261 (Px64) and H.263: Image Preparation, Coding Algorithms, Data Stream, H.263+ and H.263L, MPEG: Video Encoding, Audio Coding, Data Stream, MPEG-2, MPEG-4, MPEG-7, Fractal Compression.

Unit III

Optical Storage Media: History of Optical Storage, Basic Technology, Video Discs and Other WORMs, Compact Disc Digital Audio, Compact Disc Read Only Memory, CD-ROM Extended Architecture, Further CD-ROM-Based Developments, Compact Disc Recordable, Compact Disc Magneto-Optical, Compact Disc Read/Write, Digital Versatile Disc. Content Analysis: Simple Vs.

Complex Features, Analysis of Individual Images, Analysis of Image Sequences, Audio Analysis, Applications.

Unit IV

Data and File Format Standards: Rich-Text Format, TIFF File Format, Resource Interchange File Format (RIFF), MIDI File Format, JPEG DIB File Format for Still and Motion Images, AVI Indeo File Format, MPEG Standards, TWAIN.

Unit V

Multimedia Application Design: Multimedia Application Classes, Types of Multimedia Systems, Virtual Reality Design, Components of Multimedia Systems, Organizing Multimedia Databases, Application Workflow Design Issues, Distributed Application Design Issues.

Text Books:

1. Ralf Steinmetz, Klara Narstedt: Multimedia Fundamentals: Vol 1-Media Coding and Content Processing, First Edition, PHI, 2010.
2. Prabhat K. Andleigh, Kiran Thakrar: Multimedia Systems Design,1st Edition, PHI, 2011.

Reference Books:

1. K.R. Rao, Zoran S. Bojkovic and Dragorad A. Milovanovic: Multimedia Communication Systems: Techniques, Standards, and Networks, 1st Edition, PHI, 2010.
2. Nalin K Sharad: Multimedia information Networking, PHI, 2002.

Course Outcomes (COs):

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

1. Identify the basic concepts of media, data streams and audio technology. (PO-1, PSO-1)
2. Implement different data compression techniques including video, audio and fractal compression. (PO-1,2, PSO-2)
3. Demonstrate different optical storage media including content Analysis. (PO-1,2,3,4, PSO2)
4. Identify the different data and file format standards like TIFF, RIFFF, MIDI and MPEG. (PO-1,3, PSO-1)
5. Analyze multimedia application design methods like Virtual Reality design and workflow design. (PO-1,2,3,4,5, PSO-2)

Blockchain Essentials & DApps

Course Code: CSE746 **Credits: 3:0:0**
Prerequisites: **Contact Hours: 42**
Course Coordinator/s: Dr. Monica R Mundada

Course Contents

Unit I

Distributed systems, CAP theorem, Byzantine Generals problem, Consensus. The history of blockchain, Introduction to blockchain, various technical definitions of blockchains, Generic elements of a blockchain, Features of a blockchain, Applications of blockchain technology, Tiers of blockchain technology, Consensus in blockchain, CAP theorem and blockchain, Benefits and limitations of blockchain.

Unit II

Decentralization using blockchain, Methods of decentralization, Blockchain and full ecosystem decentralization, Smart contract, Decentralized organizations, decentralized autonomous organizations, decentralized autonomous corporations, Decentralized autonomous societies Decentralized applications, Platforms for decentralization.
Cryptographic primitives: Symmetric cryptography, Asymmetric cryptography, Public and private keys Hash functions: Compression of arbitrary messages into fixed length digest, Easy to compute, Pre-image resistance, Second pre-image resistance, Collision resistance, Message Digest (MD), Secure Hash Algorithms (SHAs), Merkle trees, Patricia trees, Distributed hash tables (DHTs), Digital signatures, Elliptic Curve Digital signature algorithm (ECDSA).

Unit III

Bitcoin, Bitcoin definition, Transactions, The transaction life cycle, The transaction structure, Types of transaction, The structure of a block, The structure of a block header, The genesis block, The bitcoin network, Wallets, Smart Contracts-History, Definition, Ricardian contracts, Smart contract templates, Oracles, Smart Oracles, Deploying smart contracts on a blockchain, The DAO.

Unit IV

Ethereum 101, Introduction, Ethereum clients and releases, The Ethereum stack, Ethereum blockchain, Currency (ETH and ETC), Forks, Gas, The consensus mechanism, The world state, Transactions, Contract creation transaction, Message call transaction, Elements of the Ethereum blockchain, Ethereum virtual machine (EVM), Accounts, Block, Ether, Messages, Mining, The Ethereum network Hands-on: Clients and wallets – Geth.

Unit V

Hyperledger: Reference architecture Requirements Fabric: Hyperledger Fabric, Fabric architecture, Membership services, Blockchain services, Components of the Fabric, Sawtooth lake: PoET, Transaction families, Consensus in Sawtooth, Development environment, Corda: Architecture, State objects, Transactions Consensus, Flows, Components, Nodes, Permissioning service, Network map service, Notary service, Oracle service Transactions, Vaults, CorDapp.

Text Book:

1. Imran Bashir. "Mastring BlockChain", Packt.

Reference Book:

1. Mastering Bitcoin: Programming the Open Blockchain Paperback – 2017 By Andreas M. O'rielly.

Course Outcomes (COs):

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

1. Illustrate the Blockchain terminologies with its applications. (PO-1,2,3, PSO-1,2)
2. Analyse the working principles of Blockchain. (PO-1,2,3, 6, PSO-1,2,3)
3. Comprehend the principles & methodologies used in Bitcoin and able to deploy smart contract. (PO-1,2,3, PSO-1,2)
4. Create Ethereum Network, Wallets, Nodes, Smart contract & Dapps (PO-1,2,3,4,5,6, PSO-1,2)
5. Develop Blockchain Based Application Architecture using Hyperledger. (PO-1,2,3,4,5,6, PSO-1,2)

Distributed Systems

Course Code: CSE751 **Credits: 3:0:0**
Prerequisites: OS **Contact Hours: 42**
Course Coordinator/s: Dr. Sini Anna Alex

Course Contents

Unit I

Introduction: Definition, Relation to computer system components, Motivation, Primitives for distributed communication, Synchronous versus asynchronous executions, Design issues and challenges.

A model of distributed computations: A distributed program, A model of distributed executions, Models of communication networks, Global state of a distributed system, Cuts of a distributed computation, Past and future cones of an event.

Logical time: Introduction, A framework for a system of logical clocks, Scalar time, Vector time, efficient implementations of vector clocks, Jard–Jourdan's adaptive technique. Relation to parallel multiprocessor/multicomputer systems, Message-passing systems versus shared memory systems.

Unit II

Global state and snapshot recording algorithms: Introduction, System model and definitions, Snapshot algorithms for FIFO channels, Variations of the Chandy–Lamport algorithm, Snapshot algorithms for non-FIFO channels, Snapshots in a causal delivery system, Monitoring global state.

Terminology and basic algorithms: Topology abstraction and overlays, Classifications and basic concepts, Synchronizers, Maximal independent set (MIS), Leader election. Complexity measures and metrics, Program structure, Elementary graph algorithms.

Unit III

Message ordering and group communication: Message ordering paradigms, Asynchronous execution with synchronous communication, Synchronous program order on an asynchronous system, Group communication, Causal order (CO), Total order, Classification of application-level multicast algorithms.

Termination detection: Introduction, System model of a distributed computation, Termination detection using distributed snapshots, Termination Detection by weight throwing, a spanning-tree based termination detection algorithm, Termination detection in a very general distributed computing model, Termination detection in the atomic computation model. A nomenclature for multicast, Propagation trees for multicast.

Unit IV

Distributed mutual exclusion algorithms: Introduction, Preliminaries, Lamport's algorithm, Ricart–Agrawala algorithm, Singhal's dynamic information-structure algorithm, Lodha and Kshemkalyani's fair mutual exclusion algorithm, Quorum-based mutual exclusion algorithms, Maekawa's algorithm, Agarwal–El Abbadi quorum-based algorithm, Token-based algorithms, Raymond's tree-based algorithm.

Deadlock detection in distributed systems: Introduction, System model, Preliminaries, Mitchell and Merritt's algorithm for the single resource model, Chandy–Misra–Haas algorithm for the AND model, Chandy–Misra–Haas algorithm for the OR model, Kshemkalyani–Singhal algorithm for the P-out-of- Q model. Models of deadlocks, Knapp's classification of distributed deadlock detection algorithms.

Unit V

Global predicate detection: Stable and unstable predicates, Modalities on predicates, Centralized algorithm for relational predicates, Conjunctive predicates, Distributed algorithms for conjunctive predicates.

Consensus and agreement algorithms: Problem definition, Overview of results, Agreement in a failure-free system (synchronous or asynchronous), Agreement in (message-passing) synchronous systems with failures, Agreement in asynchronous message-passing systems with failures.

Peer-to-peer computing and overlay graphs: Introduction, Data indexing and overlays, unstructured overlays, Chord distributed hash table. Graph structures of complex networks, Scale-free networks.

Text Book:

1. Ajay D. Kshemkalyani, and Mukesh Singhal "Distributed Computing: Principles, Algorithms, and Systems", Cambridge University Press, 2008 (Reprint 2013).

Reference Books:

1. John F. Buford, Heather Yu, and Eng K. Lua, "P2P Networking and Applications", Morgan Kaufmann, 2009 Elsevier Inc.
2. Kai Hwang, Geoffrey C. Fox, and Jack J. Dongarra, "Distributed and Cloud Computing: From Parallel processing to the Internet of Things", Morgan Kaufmann, 2012 Elsevier Inc.

Course Outcomes (COs):

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

1. Identify the design issues and Challenges in building distributed systems. (PO-1, 3, 4, PSO-1, 3)
2. Explore basic distributed graph algorithms, synchronizers, and recording global state of distributed computation. (PO-3, 4, 9, PSO-1)
3. Analyze ways to achieve various message ordering schemes for detecting termination of a distributed computation. (PO-1, 3, 4, PSO-1, 3)
4. Discuss distributed algorithms to implement Mutual Exclusion and Deadlock detection. (PO-1, 3, 4, 9, PSO-1, 2)
5. Identify Consensus and agreement algorithms and P2P overlay problems (PO-3, 9, PSO-1, 2, 3)

Cyber Physical Systems

Course Code: CSE752 **Credits: 3:0:0**
Prerequisites: **Contact Hours: 42**
Course Coordinator/s: Dr. Anita Kanavalli

Course Contents

Unit I

Modeling, design, analysis, and implementation of cyber-physical systems

Unit II

Dynamic behavior modeling-Continuous Dynamics: Newtonian Mechanics, Actor Models, Properties of Systems ,Feedback Control, Discrete Dynamics : Discrete Systems, The Notion of State, Finite-State Machines, Extended State Machines, Nondeterminism, Behaviors and Traces, Hybrid Systems, Modal Models Classes of Hybrid Systems .State machine composition: Composition of State Machines Concurrent Composition, Hierarchical State Machines Concurrent Models of Computation Structure of Models, Synchronous-Reactive Models, Dataflow Models of Computation, Timed Models of Computation.

Unit III

Sensors and actuators Models of Sensors and Actuators, Common Sensors, Actuators. Embedded systems and networks Types of Processors, Parallelism

Unit IV

Feedback control systems- Analysis and verification techniques, temporal logic, and model checking- Invariants and Temporal Logic Invariants, Linear Temporal Logic. Reachability Analysis and Model Checking: Open and Closed Systems Reachability Analysis. Abstraction in Model Checking, Model Checking Liveness Properties.

Unit V

Machine learning topics: Introduction to ML Supervised, Unsupervised, Reinforcement Frameworks, Introduction to Matlab / Python / Numpy, Preprocessing and Dimensionality Reduction, Regression, Classification Algorithms.

Reference Books/Web Links:

1. Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, Second Edition, http://LeeSeshia.org, ISBN 978-1-312-42740-2, 2015.
2. Machine Learning Algorithms and Applicationshttps://www.researchgate.net/publication/303806260_Machine_Learning_Algorithms_and_Applications

Course Outcomes (COs):

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

1. Apply cyber-physical systems fundamentals to application domains such as connected and autonomous vehicles, industrial internet, and smart and connected health
2. Implement cyber-physical systems solutions (e.g., embedded networking protocols, real-time scheduling algorithms, and networked control algorithms).
3. Explore (e.g., survey) cutting-edge research findings in cyber physical systems.
4. Apply feedback control on various open and closed systems
5. Apply ML techniques on cyber physical systems

Software Testing

Course Code: CSE753 **Credits: 3:0:0**
Prerequisites: **Contact Hours: 42**
Course Coordinator/s: Dr. T N R Kumar

Course Contents

Unit I

A Perspective on Testing, Examples: Basic definitions, Test cases, Insights from a Venn diagram, Identifying test cases, Error and fault taxonomies, Levels of testing. Examples: Generalized pseudo code, the triangle problem, The Next Date function, the commission problem, The SATM (Simple Automatic Teller Machine) problem, the currency converter, Saturn windshield wiper.

Unit II

Boundary Value Testing, Equivalence Class Testing, Decision Table-Based Testing: Boundary value analysis, Robustness testing, Worst-case testing, Special value testing, Examples, Random testing, Equivalence classes, Equivalence test cases for the triangle problem, Next Date function, and the commission problem, Guidelines and observations. Decision tables, Test cases for the triangle problem, Next Date function, and the commission problem, Guidelines and observations. Path Testing, Data Flow Testing: DD paths, Test coverage metrics, Basis path testing, guidelines and observations, Definition-Use testing, Slice-based testing, Guidelines and observations.

Unit III

Levels of Testing, Integration Testing: Traditional view of testing levels, Alternative life-cycle models, The SATM system, separating integration and system testing. A closer look at the SATM system, Decomposition-based, call graph-based, Path-based integrations, System Testing, Interaction Testing: Threads, Basic concepts for requirements specification, finding threads, Structural strategies and functional strategies for thread testing, SATM test threads, System testing guidelines, ASF (Atomic System Functions) testing example. Context of interaction, A taxonomy of interactions, Interaction, composition, and determinism, Client/Server Testing.

Unit IV

Process Framework: Validation and verification, Degrees of freedom, Varieties of software. Basic principles: Sensitivity, redundancy, restriction, partition, visibility, Feedback. The quality process, Planning and monitoring, Quality goals, Dependability properties, Analysis, Testing, Improving the process,

Organizational factors, Fault-Based Testing, Test Execution: Overview, Assumptions in fault-based testing, Mutation analysis, Fault-based adequacy criteria, Variations on mutation analysis. Test Execution: Overview, from test case specifications to test cases, Scaffolding, Generic versus specific scaffolding, Test oracles, Self-checks as oracles, Capture and replay.

Unit V

Planning and Monitoring the Process, Documenting Analysis and Test: Quality and process, Test and analysis strategies and plans, Risk planning, Monitoring the process, Improving the process, The quality team, Organizing documents, Test strategy document, Analysis and test plan, Test design specifications documents, Test and analysis reports.

Text Books:

1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Auerbach Publications, 2012.
2. Mauro Pezze, Michal Young: Software Testing and Analysis –Process, Principles and Techniques, 1st Edition, Wiley India, 2011.

Reference Books:

1. Aditya P Mathur: Foundations of Software Testing, 1st Edition, Pearson Education, 2008.
2. Srinivasan Desikan, Gopalaswamy Ramesh: Software testing Principles and Practices, 2nd Edition, Pearson Education, 2007.

Course Outcomes (COs):

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

1. Identify test cases, Error and fault taxonomies, Levels of testing. (PO-1, 2, 12, PSO-2, 3)
2. Classify different types of testing (Boundary Value Testing, Equivalence Class Testing and Decision Table-Based Testing). (PO-1, 2, 5, 12, PSO-2, 3)
3. Recognize Alternative life - cycle models, recognize Basic concepts for requirements specification, assess context of interaction. (PO-1, 2, 3, 5, 12, PSO-2, 3)
4. Recognize approaches for Test Execution: from test case specifications to test cases, Scaffolding, Generic versus specific scaffolding. (PO-1, 2, 5, 12, PSO-2, 3)
5. Identify and plan strategies to test design specifications document. (PO-1, 2, 3, 5, 12, PSO-2, 3)

Introduction to Deep Learning

Course Code: CSE754 **Credits: 3:0:0**
Prerequisites: **Contact Hours: 42**
Course Coordinator/s: Dr. Sangeetha J

Course Contents

UNIT I

Introduction: What is a Neural Network?, The Human Brain, Models of a Neuron, Neural Networks Viewed As Directed Graphs, Feedback, Network Architectures, Rosenblatt's Perceptron: Introduction, Perceptron, The Perceptron Convergence Theorem, Relation Between the Perceptron and Bayes Classifier for a Gaussian Environment. (Text Book 1 – Introduction 1, Chapter 1).

UNIT II

Multilayer Perceptrons: Introduction, Batch Learning and On-Line Learning, The Back-Propagation Algorithm, XOR Problem, Heuristics for Making the Back-Propagation Algorithm Perform Better, Back Propagation and Differentiation. (Text Book 1 - Chapter 4).

UNIT III

Regularization for Deep Learning: Parameter Norm Penalties - L2 Parameter Regularization, Dataset Augmentation, Semi-Supervised Learning, Multi-Task Learning, Early Stopping, Parameter Tying and Parameter Sharing, Sparse Representations, Dropout, Adversarial Training.

Optimization for Training Deep Models: Challenges in Neural Network Optimization – Ill Conditioning, Local Minima, Plateaus, Saddle Points and Other Flat Regions. Cliffs and Exploding Gradients, Basic Algorithms, Algorithms with Adaptive Learning Rates (Text Book 2 - Chapter 7 and Chapter 8).

UNIT IV

Convolution neural networks: The Convolution Operation, Motivation, Pooling, Convolution and Pooling as an Infinitely Strong Prior, Variants of the Basic Convolution Function, Structured Outputs, Data Types, Efficient Convolution Algorithms, Convolutional Networks and the History of Deep Learning, Text Book 2 - Chapter 9).

UNIT V

Sequence Modeling: Recurrent and Recursive Nets: Unfolding Computational Graphs, Recurrent Neural Networks, Bidirectional RNNs, Encoder-Decoder Sequence-to-Sequence Architectures, Deep Recurrent Networks, Recursive Neural Networks, The Long Short-Term Memory and Other Gated RNNs (Text Book 2 - Chapter 10).

Text Books:

1. Simon Haykin, Neural networks and Learning Machines, Third Edition, Pearson, 2016
2. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016.

Reference Book:

1. Neural Networks and Deep Learning by Michael Nielsen http://neuralnetworksanddeeplearning.com/

Course Outcomes (COs):

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

1. Illustrate the concepts and applications of neural networks and deep learning. (PO - 1, 2, 3, 4, 5, 10, 12, PSO - 2, 3)
2. Illustrate various types of learning work and its applications. (PO - 1, 2, 3, 4, 5, 10, 12, PSO - 2, 3)
3. Illustrate the Regularization and Optimization for Deep learning. (PO - 1, 2, 3, 4, 5, 10, 12, PSO - 2, 3)
4. Apply deep feed forward networks and convolutional to solve practical problems. (PO - 1, 2, 3, 4, 5, 10, 12, PSO - 2, 3)
5. Demonstrate recurrent, recursive nets function and practical problems can be mapped to these functions. (PO - 1, 2, 3, 4, 5, 10, 12, PSO - 2, 3)

Soft Computing

Course Code: CSE756 **Credits: 3:0:0**
Prerequisites: **Contact Hours: 42**
Course Coordinator/s: Dr. Jagadish S Kallimani

Course Contents

Unit I

Introduction: Neural networks, Fuzzy logic, Genetic algorithms, Hybrid systems, Artificial Neural Networks: Fundamental concept, Evolution, Basic model of ANN, Important terminologies of ANN, MP neuron, Hebb Network.

Unit II

Supervised Learning Network: Perceptron Networks, Adaptive linear neuron, multiple adaptive linear neurons, Back propagation Network.

Unit III

Introduction to Fuzzy logic, classical sets and fuzzy sets: Classical sets, Fuzzy sets. Classical relations and fuzzy relations: Cartesian product of relation, Classical relation, Fuzzy relations, Tolerance and equivalence relations. Membership functions: Features, Fuzzification, methods of membership value assignments.

Unit IV

Defuzzification: Lambda-cuts for fuzzy sets, Lambda-cuts for fuzzy relations, Defuzzification methods. Fuzzy decision making: Individual, multi person, multi objective, multi attribute, and fuzzy Bayesian decision making.

Unit V

Genetic algorithms: Introduction, Basic operations, Traditional algorithms, Simple GA, General genetic algorithms, the schema theorem, Genetic programming, applications.

Text Book:

1. Principles of Soft computing, S N Sivanandam, Deepa S. N, Wiley, India, (Chapters 1, 2, 3(Up to 3.5), 7, 8, 9, 10, 13, 15 (up to 15.6 & 15.9,15,10).

Reference Book:

1. Neuro-fuzzy and soft computing, J.S.R. Jang, C T Sun, E Mizutani, PHI (EEE edition) ISBN: 978-81-203-2243-1.

Course Outcomes (COs):

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

1. Describe various problems on artificial neural networks and familiarize all concepts on various networks and applications in them. (PO-1,2,11, PSO- 2)
2. Identify the compositions of neural networks, perceptrons and other networks. (PO-5,11, PSO-2)
3. Examine various fuzzification techniques and practice them. (PO-1,2, 4,11, PSO-2)
4. Design problems and obtain crisp values from fuzzy data using defuzzification. (PO-2,5,11, PSO-2)
5. Summarize various compositions and complexities of genetic algorithms. (PO- 1,4,11, PSO-2)

Big Data Laboratory

Course Code: CSL76 **Credits: 0:0:1**
Prerequisites: **Contact Hours: 28**
Course Coordinator/s: Dr. Ganeshayya Shidaganti

Course Contents

1. Downloading and Installing Hadoop.
2. Understanding Hadoop Distributed File system (HDFS); Explore Basic Hadoop Commands; Different Hadoop Modes. Startup Scripts, Configuration files.
3. Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.
4. Understanding Anatomy of MapReduce Job Run and Workflows and developing various MapReduce Applications.
5. Downloading and Installing Apache Spark.
6. Running Apache Spark Applications using Scala/ Python.
7. Downloading and Installing Apache Pig.
8. Developing and Testing Pig Latin Scripts for Sort, Group, Join, Project, and Filter your Data.
9. Downloading and Installing Apache Hive.
10. Understanding Hive Shell and Manipulating HiveQL Quires for Create, Alter, and Drop Databases, Tables, Views, Functions, and Indexes.

References:

1. Tom White “Hadoop: The Definitive Guide” Third Edition, O’reilly Media, 2012.
2. Ryza, Sandy, Uri Laserson, Sean Owen, and Josh Wills. Advanced analytics with spark: patterns for learning from data at scale. " O'Reilly Media, Inc.", 2017.
3. Gates, Alan, and Daniel Dai. Programming pig: Dataflow scripting with hadoop. "O'Reilly Media, Inc.", 2016.
4. Capriolo, Edward, Dean Wampler, and Jason Rutherglen. Programming Hive: Data warehouse and query language for Hadoop. " O'Reilly Media, Inc.", 2012.Mining Concepts and Techniques”, 2nd Edition, Elsevier.

Course Outcomes (COs):

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

1. Configure Hadoop and Perform File Management Tasks (PO-1, 2, 3,4,5 PSO-3)
2. Explore the Hadoop Ecosystem and MapReduce Programming to perform various Business Applications associated with Big Data Analytics. (PO-1, 2, 3,4,5 PSO-3)
3. Analyze Big Data Applications Using Pig, Hive and Spark. (PO-1, 2, 3,4,5 PSO-3)

Micro Services laboratory

Course Code: CSL77 **Credits: 0:0:1**
Prerequisites: Java Programming **Contact Hours: 28**
Course Coordinator: Dr. Geetha J

Course Contents

- Overview of Microservices
- Microservice Architecture
- Preparing the Environment
- Introduction to Spring Boot for Microservices
- Getting started with Hello World Microservice
- Building microservices with Spring Boot
- Testing microservices
- Communication between 2 or more microservices through REST services (setting up Eureka service and two or three services)
- Distributed log tracing by using zepkln and spring cloud sleuth. (i.e. tracing a request from multiple microservice)
- Mocking and Testing service by using Mocikto and Junit
- Connecting spring microservices with a data base.
- Setting up Swagger ui for a microservice.
- Creating custom annotations (at method level and class level)
- Deploying microservices using Docker/Kubernetes

Reference Books:

1. Chris Richardson: Microservices Patterns with Examples in Java, Manning Publications Co., First Edition, 2019.
2. Moises Macero: Learn Microservices with Spring Boot: A Practical Approach to RESTful Services using RabbitMQ, Eureka, Ribbon, Zuul and Cucumber, A Press, First Edition, 2017.
3. Sourabh Sharma: Mastering Microservices with Java 9, Packt Publishing Ltd, Second Edition, 2017.

Course Outcomes (COs):

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

1. Illustrate the importance of Microservices as an Architecture Implementation. (PO-1,2,3, PSO-2)
2. Develop an application on Java with MongoDB and convert over monolithic structure to Micro services. (PO-1,2,3, PSO-3)
3. Deploy application on docker and Access the Kubernetes. (PO-1,2,3,5, PSO-3)

Seminar

Course Code: CSSE **Credits: 0:0:1**
Prerequisites: **Contact Hours: 28**
Course Coordinator: Dr. Sini Anna Alex

Seminars are used as course delivery modes to encourage students to gather current trends in technology, research literature, and self-learn topics of their interest. Seminars require students to research a technical topic, make presentations and write a detailed document on their findings individually under the guidance of faculty.

The student is expected to:

1. Identify seminar topics based on contemporary technical, societal and environmental issues.
2. Conduct literature survey on complex issues in the selected domain
3. Explore advanced technologies
4. Make good oral and written technical presentations

Course outcomes (COs):

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

1. Identify seminar topics based on contemporary technical, societal and environmental issues. (PO-2,4, 6,7,8,11, PSO-1,2)
2. Conduct a literature survey on complex issues in the selected domain. (PO-2, 4, 6, 7, 8,11, PSO-1,2)
3. Explore advanced technologies. (PO-3, 11,12, PSO-1,2)
4. Create good oral and written technical presentations. (PO-3, 4, 5, 10, PSO-1)
5. Discuss the outcome of the selected domain and use recent technology for creating reports (PO-3, 5, 8, 9,11, PSO-1)

Rubrics for Seminar Evaluation:

The rubrics for Seminar evaluation is provided in the project work book.

Internship

Course Code: CSIN **Credits: 3:0:0**
Prerequisites: Nil
Course Coordinator: Dr. S. Rajarajeswari

Guidelines:

- The student can do the Internship during the summer vacation after the 2^{nd},4^{th},6^{th} semester or in 8^{th} semester.
- The student should take prior permission from the department committee before carrying out the internship. The offer letter should be submitted to the Dept. coordinator.
- The duration of the Internship is 3- 4 -6 weeks after each semester (total Six months).
- The Internship assessment form assessed by the company mentor, report of the Internship, Internship completion certificate, needs to be submitted to the department coordinator after completion of the Internship.
- The department will constitute a committee for the evaluation of Internship of student and based on the following rubrics, the student will be evaluated.

Deliverables for Student Performance in Internship:

Internship Title:

Company

Name:

Name of Student:

Name of Supervisor at

Company: Name of Supervisor

at College:

Internship Assessment – Rubrics

Each supervisor must fill a rubric for each student:

Tools and new Technology Learnt	**Basic (0-4 Pts)** Few sources at the Industry, aware of quality of resources and relevance to tools and Techniques at hand	**Good (5-7 Pts)** Multiple sources of high quality, good judgment of the information, identification of gaps in knowledge at the Industry and Academics.	**Very Good (10 Pts)** Multiple sources of high quality, well researched and analyzed, continuous efforts at acquiring Information. Identification of the application of the tools and Technology learnt to the present market.	**Total**
Relevance of the topic chosen to the current market	Fairly Relevant	Moderately Relevant	Highly Relevant	

Rubrics for the Report Writing, Demo and Presentation are maintained in a separate workbook

Course Outcomes (COs):

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

1. Schedule milestones of deliverables and formulate the requirements of the proposed work. (PO-2,9,11, PSO-1,2)
2. Apply the engineering knowledge to develop software in an industry setting. (PO-1,2,3,5, PSO-1,2)
3. Develop the inter-personal skills required to work in a professional team. (PO-9, 10, 11, PSO-2, 3)
4. Engage in independent study of technology required for development of software. (PO-12, PSO-2, 3)
5. Demonstrate the project and appraise its effectiveness. (PO-10, PSO-3)

Project Work

Course Code: CSP **Credits: 0:0:14**
Prerequisites: Nil
Course Coordinator: Dr. S. Rajarajeswari

Course Contents

As a part of project, all the eligible final year students must carry out the following activities:

1. Students should form a group to carry out their project. The minimum group size is 2 and maximum group size is 4.
2. The groups will be attached to one Internal Guide (and Co-guide if necessary) by the Department.
3. Students can carry out their project in-house or in a reputed organization (to be approved by Internal Guide and HOD).
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 Project Report (plagiarism should be less than 10%) and demonstrating their work.
10. Publish the Project work in a Scopus indexed Conference/Journal with quartile ranking (Q1, Q2, Q3) IEEE code of ethics for authors should be strictly adhered.
11. The Continuous Internal Evaluation is based on a presentation and report for 100 marks scaled down to 50 marks.
    - The evaluation will be done by the internal guide and a co- examiner twice during the semester.
        - Weekly report to the Guide (10 marks)
        - Mid-semester evaluation: Students must do a group presentation and produce documents of system requirements, and system design (during 6^{th} week) (40 marks)
        - Final Evaluation: At the End of the semester students must do a group presentation, demonstrate the project work and submit the complete report. (during 13^{th} week) (50 marks)
12. The SEE Project Viva Voce will be conducted for 50 marks with an Internal and External Examiner.

General Rubrics for Project Evaluation:

The rubrics for project evaluation are provided as per the below tables.

Criteria for Evaluation	Level A(10) 90-100	Level (8) 75-90	Level C(6) 50-75	Level D(5) Up to 50
Data Elicitation Phase	Has investigated new trends in their area of interest, Review the challenges in That area. Data elicitation should Include new concepts.	Has investigated new trends in their area of interest, Review some challenges in that area	Has investigated new trends in their Area of interest.	Has not investigated much on new trends in their area of interest
Problem Definition	Has investigated problem domain extensively	Problem domain well understood, clear and specific description of problem, relevance well identified	Moderately awareness of problem domain, clear description, broad idea about relevance to current technical and social context	Minimal awareness of problem domain, Vague description, little idea about relevance to current Technical and social context
Planning	Precise Schedule and Effort Estimation using Tools.	Precise Schedule and Effort Estimation Manually.	Schedule and Effort Estimation.	Inappropriate Schedule and Effort Estimation.
Project management	Has taken leadership role in the project and monitored progress of the project. In addition, has completed all tasks assigned to him.	Has monitored the progress of the project and completed all tasks assigned to him.	Has completed all the tasks assigned to him.	Has not completed all the tasks assigned to him.
Literature Survey	Has read more than 10 specific papers from reputed journals or 20 papers from conference and 3 books in the area of the project.	Has read more than 7 papers from reputed journals or 12 papers from conference and 2 books in the area of the project.	Has read more than 5 papers from reputed journals or 8 papers from conference and 1 books in the area of the project.	Minimal, mostly from general sources, without focused study.
Requirements Specification	Complete functional, Non-functional, Performance, Security related,	Partial functional, Non-functional, Performance, Security	In-Complete functional, Non-functional, Performance, Security related,	Few, narrow and incomplete requirements

	Clear and Measurable (in terms of SMART Matrix)	related, Clear and Measurable (in terms of SMART Matrix)	Clear and Measurable (in terms of SMART Matrix)	
System Design	The team has to play a role of main Architect in the project , Designed all the components of the project	The team has to play the role of main Architect in the project , Designed 90% of the components of the project	The team has to play the role of main Architect in the project , Designed 75% of the components of the project	The team has to play the role of main Architect in the project, Designed 50% of the components of the project
Implementation	Has decided the relevant tools and platforms required for the project by evaluating alternatives. coded all the components designed by him by following the standard coding guidelines.	Has coded all the components designed by him by following the Standard coding guidelines.	Has coded all the components designed by him by not following the standard coding guidelines.	Has not coded all the components designed by him by not following the standard coding guidelines.
Testing and Results	Meets all the requirements, Optimized Solution, Proper Test Plan, Has performed Integration Testing, Performance Testing	Barely meets all the requirements, Not Optimized Solution, Poor Test Plan, Has not performed Integration Testing, Performance Testing	Barely meets all the requirements, Not Optimized Solution, Poor Test Plan, Has not performed Integration Testing, Performance Testing	Haphazard testing, barely meets requirements, unable to infer results.
Report Writing	Excellent Organization,	Good Organization, No technical or Grammar errors, Concise and Precise, Incomplete documentation, done on Latex	Average Organization, No technical or Grammar errors, Concise and Precise, Incomplete documentation, Not done on Latex	Poor Clarity in technical contents and organization, error in grammar, not done in Latex

Presentation and Viva-voce	Excellent Professional and Technical communication, Effective Presentations, able to analyze technically and clarify views in viva-voce	Good Professional and Technical communication , Effective Presentations.	Average Professional and Technical communication, Effective Presentations, Unable to analyze technically and Clarify views in viva-voce	Poor Technical communication, Not an Effective Presentations, Unable to analyze technically and clarify views in viva-voce

Rubrics for Final year Project Paper (IEEE Format)			
Criteria and Qualities	**Poor (1-4)**	**Adequate (5-7)**	**Good (8-10)**
Abstract			
Content & Length	The abstract does not cover each sectionof the paper; conclusion do not match study findings The abstract exceeds the word limit (250 words)	The abstract covers most sections of the paper The abstract is an appropriate length	The abstract covers all sections of the paper and coveys the key findings from the study in a clear way The abstract is an appropriatelength
Introduction			
Literature Review	Literature reviewed has weak or no connection to the topic under study. A clear rationale for the study aim/ purpose (the gap in the literature) is not identified. Major sections of pertinent literature are omitted or literature reviewed is not from scholarly sources	Literature reviewed relatesto the study topic. All major sections of the pertinent literature are included, but may not be covered in depth or a few areas of pertinent literature are not covered. Most articles/ sources reviewed are from scholarly sources but a few are not.	Literature reviewed relates to themain topic and sets up the rationale for the study aim/ purpose by clearly identifying a gap in the literature. Introduction covers relevant andcurrent articles in detail. Only scholarly articles are used to build the argument for the need for the study.
Study Aim/ Purpose	Neither implicit nor explicit reference is made to the study aim or purpose orthe study aim appears unrelated to the literature reviewed.	Readers are aware of the overall problem or aim ofthe study but the aim is not clear or the study aimis not clearly linked to a gap in the literature	The aim clearly flows from the groundwork laid/ literature reviewed and a clear case is madefor the need for the study aims based on a gap in the literature.

Method	Critical details necessary to understandhow the study was conducted are lacking. Sections such as study design, procedures, measures/ instruments, analytic approach are missing.	Study methods are generally described but information regarding nuances of how the studywas conducted is more limited.	The study design is described indetail. If quantitative data were collected, psychometric properties of instruments are provided (e.g. validity and reliability) are provided. The sampling strategy (inclusion/ exclusion criteria), study settingand data collection procedures are described in detail. The approach used for data analysesare described.
Results			
Presentation of Data/ Findings	Results are poorly Described and/or do not align withdescription of study methods. The Results section includes "conclusions"instead of simply presenting the data and/ or analyses.	Results are adequately described and aligned with description of study methods. Analyses of data are completed correctly.	Results/products/outcomes are described in detail and align withdescription of study methods. Analyses of data are sophisticatedand precise
Tables and Figures	Tables/Figures are missing and/or unlabeled. If present, they do not clearly present the study findings/data and/ or are redundant with one another.	Tables/Figures are presentin the paper and are labeled. Tables/ Figures adequately present data/ findings but may be redundant with data presented in the text	Tables/Figures are present and are labeled. Tables and Figuresprovide critical information and are organized in such a way as toenhance understanding of the study results.

Course Outcomes (COs):

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

1. Review the current state of Art and trends in their area of interest and identify a suitable problem in their chosen subject domain with justification. (PO-1,2,6, 7, 9, 10, 11, PSO-2,3)
2. Survey the available research literature/documents for the tools and techniques to be used. (PO-1, 2, 5, 8, 9, 10, 11, 12, PSO-2,3)
3. Examine the functional, non-functional, and performance requirements of their chosen problem definition. (PO-1,2,4, 9, 10, 11, 12, PSO-2,3)
4. Design system architecture and different components and develop all the system components using appropriate tools and techniques. (PO-1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, PSO-2,3)
5. Work effectively in a team and use good project management practices and defend the project work as a team (PO-5, 8, 9, 10, 11, 12,PSO-2,3)