

Curriculum Master of Science in Information Technology

(2026)



Rawalpindi Women University, Rawalpindi

Table of Contents

Rawalpindi Women University Vision.....	iii
Rawalpindi Women University Mission Statement.....	iii
Department Mission Statement.....	iii
1. Curriculum for Master of Science in Information Technology (MS (IT)).....	1
1.1 Program Vision.....	1
1.2 Program Mission.....	1
1.3 Eligibility Criteria	1
1.4 Degree Duration.....	1
1.5 Degree Completion Requirements	1
1.6 Program Objectives and Outcomes.....	1
2. Area Covered in MS (IT) Program	2
3. Scheme of Study of MS Information Technology in the light of HEC Curriculum... 2	2
4. Roadmap of MS (IT) Program	4
5. Course Outlines.....	5

Rawalpindi Women University Vision

Empowering women through education, critical thinking and innovation for a better tomorrow.

Rawalpindi Women University Mission Statement

Rawalpindi Women University aims to promote a robust learning environment with focus on humanitarian norms and cultural values where innovative research blends with effective learning to ensure a contributory role in society.

Department Mission Statement

To deliver high-quality education in information technology, inspiring women to excel as innovative professionals, researchers, and ethical contributors to the global IT industry. By promoting economic independence, sustainable practices, and lifelong learning, the department aims to prepare students for dynamic challenges in technology-driven fields

1. Curriculum for Master of Science in Information Technology (MS (IT))

1.1 Program Vision

The vision of MS (IT) program is to produce highly skilled IT professionals who can meet evolving industry demands, empower graduates for impactful roles in software, academia, and research, and contribute to Pakistan's digital growth and socio-economic development.

1.2 Program Mission

MS (IT) Program is aimed to create, expand, and disseminate IT knowledge through academics, applications, and research, to equip students with skills that address real-world technological challenges, and to positively impact society at local, national, and global levels.

1.3 Eligibility Criteria

The minimum requirements for admission in a MS (IT) degree program are:

A degree earned after sixteen years of education (M.Sc./BS-4 year's degree program, minimum 130 credit hours) in Computer Science/Information Technology/Software Engineering / Computer Engineering or equivalent qualification, earned from a recognized university after 16 years of education with at least 60% marks or CGPA of at least 2.5 (on a scale of 4.0).

Note: "Eligibility criteria for admissions can be updated according to RWU policy"

1.4 Degree Duration

The minimum duration for completion of MS (IT) degree is two years (4-regular semester) and the maximum period is four years to complete MS (IT) degree requirements.

1.5 Degree Completion Requirements

To become eligible for award of MS degree, a student must satisfy the following requirements:

- a. Passed totaling at least 32 credit hours.
- b. Must have earned CGPA (Cumulative Grade Point Average) of at least 2.5 on a scale of 4.

1.6 Program Objectives and Outcomes

The MS (IT) comprises of both course work as well as research components. There are four 'core courses' aimed at strengthening the understanding and competence of students in information technology fundamentals. The University expects its MS graduates to pursue careers either as 'IT Faculty Members' or as 'IT Managers' in the industry.

Learning Outcomes:

1. Graduates will possess advanced and up-to-date knowledge in core and emerging areas of Information Technology.
2. Graduates will demonstrate critical and creative thinking skills to analyze and solve complex IT-related problems.
3. Graduates will apply IT principles to design and implement efficient, scalable, and real-world solutions.
4. Graduates will be capable of conducting independent research and contributing to innovations in the IT field.

2. Area Covered in MS (IT) Program

Course Group	Credit Hours
Core Course	12
Mandatory Course	05
Elective	09
Research Thesis	06
Total Credit Hours	32

3. Scheme of Study of MS Information Technology in the light of HEC Curriculum

3.1 Core Courses (12 Credit Hours)

#	Code	Course Title	Credit hours
1	IT-711	Advance Information Security and Assurance	3
2	IT-712	Advance Information Technology Infrastructure	3
3	IT-713	Advance Cloud Computing	3
4	IT-714	Mathematical Methods of Information Technology	3
5	CS-741	Advanced Computer Networks	3

Note: Department can offer any four core courses according to RWU policy.

3.2 Mandatory Course (5 Credit Hours)

#	Code	Course Title	Credit hours
1	CS-711	Research Methods	3
2	UHQ-101 / ETH-103	Understanding of Quran-I/ Ethics-I	1
3	UHQ-102 / ETH-104	Understanding of Quran-II/ Ethics-II	1

3.3 Tentative Elective Courses (9 Credit Hours)

#	Code	Course Title	Credit hours
1.	IT-701	Advance Ubiquitous Computing	3
2.	IT-702	Advance Database Management System	3
3.	IT-715	Advance Wireless Sensor Networks	3
4.	IT-716	Advance Internet of Things	3
5.	IT-717	Introduction to Software Defined Networking	3
6.	CS-712	Advanced Requirement Engineering	3
7.	CS-713	Advanced Global Software Development	3
8.	CS-714	Software Quality Engineering	3
9.	CS-715	Advanced Software Project Management	3
10.	CS-721	Image Processing	3
11.	CS-722	Computer Vision	3
12.	CS-723	Deep Learning in Computer Vision	3
13.	CS-731	Advanced Artificial Intelligence	3
14.	CS-732	Deep Learning	3
15.	CS-733	Natural Language Processing	3
16.	CS-734	Machine Learning	3
17.	CS-742	Security Management and Cryptography	3
18.	CS-743	Distributed Systems	3

Note: Above mentioned is the proposed list from the pool of CS Elective courses. The electives can be offered from the above-mentioned courses but not limited to only these. Any new course can be offered/added with the approval of BoS.

3.4 Research Thesis (06 Credit Hours)

The department started MS (IT) degree containing research work of 06 credit hours according to the guidelines of HEC. Registration in MS Thesis-I is allowed provided the student has;

- Earned at least 18 credit hours.
- Passed the “Research Methodology” course.
- CGPA is equal to or more than 2.5.

4. Roadmap of MS (IT) Program

ROADMAP of MS (CS) Program					
Semester	Course Code	Course Title	Pre-Requisites	Credit Hours	Course Type
First		Core 1	Nil	3	Core Course
		Core II	Nil	3	Core Course
		Elective I	Nil	3	Elective
		Elective II	Nil	3	Elective
	UHQ-101 / ETH-103	Understanding of Quran-I/ Ethics-I	Nil	1	Mandatory
	Total			13	
Second		Core III	Nil	3	Core Course
		Core IV	Nil	3	Core Course
		Elective III	Nil	3	Elective
	UHQ-102 / ETH-104	Understanding of Quran-II/ Ethics-II	Nil	1	Mandatory
	CS-711	Research Methods	Nil	3	Mandatory
	Total			13	
Third	CS-799	Research Thesis	Nil	06	MS Thesis
Fourth					
			Total	32	

5. Course Outlines

Department: Information Technology

Course Title: Advance Information Security and Assurance

Course Code: IT-711

Credit Hours: 03

Pre-requisite: None

Course Objectives

- To introduce students to the foundational concepts of information security, its goals, and assurance techniques.
- To explore different types of threats, vulnerabilities, and controls used to safeguard information systems.
- To understand cryptographic principles and protocols for securing communication and data.
- To analyze security policies, risk management techniques, and incident response processes.
- To enable students to assess and enhance the security posture of information systems.

Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Understand core concepts and principles of information security, including confidentiality, integrity, and availability. *(C2 Understand)*
2. Identify and assess different types of security threats and vulnerabilities. *(C4 Analyze)*
3. Apply appropriate security controls and countermeasures for protecting assets. *(C3 Apply)*
4. Evaluate the effectiveness of security policies, risk management, and compliance strategies. *(C5 Evaluate)*
5. Explain cryptographic algorithms and use them to secure communication. *(C2 Understand)*
6. Develop incident response plans and business continuity strategies. *(C6 Create)*

Course Outline

Unit 1: Introduction to Information Security

- Definition and importance of information security
- Security goals: Confidentiality, Integrity, Availability (CIA triad)
- Threats, vulnerabilities, and risk
- Security policies, procedures, and awareness
- Security governance and compliance (ISO 27001, NIST)

Unit 2: Cryptography and Secure Communication

- Basics of cryptography: symmetric and asymmetric encryption
- Block ciphers (AES, DES), RSA, Diffie-Hellman
- Hashing functions: SHA-2, SHA-3, MD5
- Digital signatures, certificates, and Public Key Infrastructure (PKI)
- SSL/TLS and secure communication protocols

Unit 3: Access Control and Authentication

- Identification, authentication, and authorization
- Access control models: DAC, MAC, RBAC, ABAC

- Biometrics, smart cards, OTP, MFA
- Directory services and single sign-on (SSO)
- Identity and Access Management (IAM)

Unit 4: Network and System Security

- Network security fundamentals (firewalls, IDS/IPS, VPNs)
- Securing operating systems and applications
- Wireless network security (WPA3, EAP)
- Endpoint protection and anti-malware solutions
- Secure software development (DevSecOps, SDLC)

Unit 5: Security Risk Management

- Risk analysis and management process
- Threat modeling and vulnerability assessment
- Penetration testing and ethical hacking basics
- Business impact analysis (BIA)
- Security audits and assurance frameworks

Unit 6: Incident Response and Disaster Recovery

- Phases of incident response
- Computer forensics basics
- Business continuity planning (BCP)
- Disaster recovery strategies and tools
- Legal, ethical, and compliance issues in incident handling

Unit 7: Security in Emerging Technologies

- Cloud security challenges and controls (CSA, shared responsibility)
- IoT security and privacy
- Security in virtualization and container environments
- Blockchain and decentralized security models
- Artificial intelligence in cybersecurity

Unit 8: Case Studies and Current Trends

- Security breaches and lessons learned
- Industry standards and certifications (CISSP, CISM, CEH)
- Data privacy laws (GDPR, HIPAA, etc.)
- Cybersecurity strategies for organizations
- Latest research directions in security and assurance

Recommended Books / Reference Material

1. Stallings, W. (2018). *Cryptography and Network Security*, 7th Edition, Pearson.
2. Whitman, M., & Mattord, H. (2021). *Principles of Information Security*, 7th Edition, Cengage Learning.
3. Pfleeger, C. P., & Pfleeger, S. L. (2015). *Security in Computing*, 5th Edition, Pearson.
4. Anderson, R. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems*, 3rd Edition, Wiley.
5. NIST Special Publications (SP800 series), ISO/IEC 27000 series.
6. Online Resources: OWASP, SANS Institute, CIS Controls, CSA Security Guidance.

Department: Information Technology
Course Title: Advance Information Technology Infrastructure
Course Code: IT-712
Credit Hours: 03
Pre-requisite: None

Course Objectives

- To provide foundational and advanced knowledge of the components and architecture of IT infrastructure.
- To understand the role of IT infrastructure in supporting business operations and enterprise systems.
- To analyze the deployment, management, and evolution of infrastructure services such as networking, computing, and storage.
- To explore cloud computing, virtualization, and data center technologies as part of modern IT infrastructure.
- To develop strategies for planning, implementing, and maintaining reliable, scalable, and secure infrastructure environments.

Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Describe the components of IT infrastructure and their roles in enterprise computing. *(C2 Understand)*
2. Evaluate different computing architectures including centralized, distributed, and cloud-based models. *(C5 Evaluate)*
3. Analyze network and storage technologies used in IT environments. *(C4 Analyze)*
4. Apply principles of virtualization and systems management to infrastructure planning. *(C3 Apply)*
5. Design and assess infrastructure for scalability, availability, and disaster recovery. *(C6 Create)*
6. Demonstrate understanding of current trends such as software-defined infrastructure and hybrid cloud. *(C2 Understand)*

Course Outline

Unit 1: Introduction to IT Infrastructure

- Definition and scope of IT infrastructure
- Evolution of infrastructure models (on-premise to hybrid/cloud)
- Components: servers, storage, networking, OS, middleware
- IT infrastructure in enterprise environments
- ITIL and service-oriented infrastructure management

Unit 2: Data Centers and Virtualization

- Data center components and architecture
- Server virtualization (Hypervisors: Type I and Type II)
- Storage virtualization and network virtualization
- Benefits and challenges of virtualization
- Virtualization tools (VMware, Hyper-V, KVM)

Unit 3: Networking in IT Infrastructure

- Network fundamentals: LAN, WAN, switching, routing

- TCP/IP stack, DNS, DHCP, NAT, VPNs
- Network security: firewalls, IDS/IPS, NAC
- Software Defined Networking (SDN) overview
- Network infrastructure management and monitoring

Unit 4: Storage and Backup Systems

- Storage architectures: DAS, NAS, SAN
- RAID levels and data redundancy
- Backup strategies and disaster recovery
- Storage management and capacity planning
- Emerging trends in storage: Object storage, S3, NVMe

Unit 5: Cloud and Hybrid Infrastructure

- Cloud service models: IaaS, PaaS, SaaS
- Cloud deployment models: public, private, hybrid, multi-cloud
- Cloud infrastructure platforms: AWS, Azure, GCP
- Virtual private clouds and container orchestration (Docker, Kubernetes)
- Cost models, SLAs, and compliance in cloud infrastructure

Unit 6: Infrastructure Security and Management

- Infrastructure security challenges and solutions
- Identity and access management (IAM)
- Patch management and configuration control
- Monitoring tools (Nagios, Zabbix, Prometheus)
- Infrastructure automation with scripts and tools (Ansible, Terraform)

Unit 7: Planning and Architecting IT Infrastructure

- Business requirements analysis
- Infrastructure lifecycle management
- High availability and failover design
- Capacity planning and scalability
- Case studies of enterprise infrastructure design

Unit 8: Trends and Future Directions

- Software-defined everything (SDx)
- Edge computing and fog computing
- Green IT and sustainable infrastructure
- IT infrastructure in smart cities and IoT ecosystems
- Infrastructure as Code (IaC) and DevOps integration

Recommended Books / Reference Material

1. Olson, D. (2020). *IT Infrastructure Architecture – Infrastructure Building Blocks and Concepts*.
2. Forouzan, B. A. (2017). *Data Communications and Networking*, 5th Edition, McGraw-Hill.
3. Erl, T., Mahmood, Z., & Puttini, R. (2013). *Cloud Computing: Concepts, Technology & Architecture*, Prentice Hall.
4. Silberschatz, A., Galvin, P. B., & Gagne, G. (2020). *Operating System Concepts*, Wiley.
5. Nelson, R., & Phillips, A. (2021). *Guide to Computer Network Security*, Springer.

6. ITIL Foundation resources (AXELOS) and online labs/documentation for AWS, Azure, VMware.

Department: Information Technology

Course Title: Advanced Database Management System

Course Code: IT-702

Credit Hours: 03

Pre-requisite: Database Management Systems (DBMS)

Course Objectives

- To provide advanced knowledge of database design, modeling, and implementation techniques.
- To explore emerging database technologies including distributed, NoSQL, and big data systems.
- To develop skills for handling complex queries, transaction management, and database optimization.
- To understand the architecture and challenges of managing large-scale data.
- To expose students to real-world use cases involving performance tuning, indexing, and secure database systems.

Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Explain advanced database concepts such as query optimization, transaction processing, and concurrency control. *(C2 Understand)*
2. Apply normalization, indexing, and optimization techniques to design efficient databases. *(C3 Apply)*
3. Analyze and manage distributed databases, NoSQL systems, and big data frameworks. *(C4 Analyze)*
4. Evaluate performance issues and suggest tuning strategies for SQL and NoSQL databases. *(C5 Evaluate)*
5. Design and implement secure and scalable data-driven applications. *(C6 Create)*
6. Demonstrate familiarity with advanced data models and emerging trends in data management. *(C2 Understand)*

Course Outline

Unit 1: Advanced Relational Database Concepts

- Extended ER model and object-relational mapping
- Relational algebra review and optimization
- SQL: advanced queries (views, triggers, stored procedures, recursive queries)
- Indexing and hashing techniques
- Query processing and evaluation

Unit 2: Database Design and Optimization

- Functional dependencies and multivalued dependencies
- Advanced normalization (BCNF, 4NF, 5NF)
- Denormalization and physical design

- Query optimization strategies
- Cost-based optimization and execution plans

Unit 3: Transaction Management

- Transaction properties (ACID)
- Concurrency control mechanisms: locking protocols, timestamp ordering
- Serializability and conflict resolution
- Recovery system: log-based recovery, checkpoints
- Deadlock handling in DBMS

Unit 4: Distributed and Parallel Databases

- Distributed DBMS architecture and design
- Data fragmentation, replication, and allocation
- Distributed query processing
- Distributed transaction management and concurrency
- Parallel databases and performance issues

Unit 5: NoSQL and Emerging Database Models

- Limitations of relational model
- Introduction to NoSQL (Key-Value, Document, Column-family, Graph databases)
- Case studies: MongoDB, Cassandra, Neo4j
- Query languages for NoSQL
- CAP theorem and BASE properties

Unit 6: Data Warehousing and OLAP

- Data warehouse architecture and schemas (star, snowflake)
- ETL processes and tools
- OLAP operations: slicing, dicing, roll-up, drill-down
- Materialized views and cube computation
- Data mining integration

Unit 7: Big Data and Cloud Databases

- Hadoop ecosystem (HDFS, MapReduce)
- Introduction to Hive, Pig, and Spark SQL
- Cloud databases: characteristics and examples (Amazon RDS, Google BigQuery)
- Scalable storage systems (HBase, DynamoDB)
- Data consistency models in the cloud

Unit 8: Security, Privacy, and Advanced Topics

- Database security challenges and access control models
- Encryption, auditing, and user privilege management
- SQL injection and mitigation
- Temporal, spatial, and multimedia databases
- Trends: Data lakes, data mesh, blockchain databases

Recommended Books / Reference Material

1. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database System Concepts*, 7th Edition, McGraw-Hill.
2. Elmasri, R., & Navathe, S. B. (2017). *Fundamentals of Database Systems*, 7th Edition, Pearson.
3. Cattell, R. (2011). *Scalable SQL and NoSQL Data Stores*, ACM Computing Surveys.
4. Harrison, G. (2015). *Next Generation Databases: NoSQL, NewSQL, and Big Data*. Apress.
5. Sadalage, P. J., & Fowler, M. (2012). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley.
6. White, T. (2015). *Hadoop: The Definitive Guide*, 4th Edition, O'Reilly Media.

Department: Information Technology

Course Title: Mathematical Methods of Information Technology

Course Code: IT-714

Credit Hours: 03

Pre-requisite: None

Course Objectives

- To provide foundational and advanced mathematical techniques required for solving problems in Information Technology.
- To enhance students' analytical skills and logical reasoning capabilities for application in IT-related domains.
- To introduce discrete mathematics, probability, linear algebra, graph theory, and optimization techniques.
- To build a mathematical framework for understanding algorithms, data structures, networks, and machine learning.
- To apply mathematical models to real-world IT problems including system modeling, decision making, and performance analysis.

Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Understand and apply discrete structures, set theory, and logic to formalize computing problems. *(C2 Understand)*
2. Apply linear algebra concepts to data representation and transformation in computing systems. *(C3 Apply)*
3. Analyze algorithms using combinatorics and probability theory. *(C4 Analyze)*
4. Utilize graph theory for modeling computer networks, trees, and relationships. *(C4 Analyze)*
5. Solve real-world IT problems using optimization and decision-making techniques. *(C5 Evaluate)*
6. Construct and interpret mathematical models for various applications in data science and systems design. *(C6 Create)*

Course Outline

Unit 1: Foundations of Discrete Mathematics

- Sets, relations, and functions
- Logic and propositional calculus
- Predicate logic and quantifiers
- Mathematical induction and recursion
- Proof techniques: direct, contradiction, contrapositive

Unit 2: Linear Algebra for IT

- Matrices and matrix operations
- Systems of linear equations
- Vector spaces, basis, and dimension
- Eigenvalues and eigenvectors
- Applications in computer graphics, machine learning, and data science

Unit 3: Combinatorics and Probability

- Counting principles, permutations, and combinations
- Binomial theorem and Pascal's triangle
- Basics of probability theory and distributions
- Conditional probability and Bayes' theorem
- Applications in algorithm analysis and information theory

Unit 4: Graph Theory and Trees

- Basic concepts of graphs, types, and representations
- Trees, spanning trees, BFS, DFS
- Shortest path algorithms (Dijkstra, Floyd-Warshall)
- Graph coloring, planarity, and connectivity
- Applications in computer networks and social networks

Unit 5: Optimization and Linear Programming

- Introduction to optimization and operations research
- Linear programming models and graphical method
- Simplex method and duality theory
- Integer programming and applications in resource allocation
- Introduction to dynamic programming and greedy algorithms

Unit 6: Advanced Topics and Applications

- Boolean algebra and switching functions
- Markov chains and queuing theory
- Fuzzy logic and its IT applications
- Information theory and entropy
- Mathematical modeling in software reliability and performance evaluation

Recommended Books / Reference Material

1. Kenneth H. Rosen (2019). *Discrete Mathematics and Its Applications*, 8th Edition, McGraw-Hill.
2. Gilbert Strang (2016). *Introduction to Linear Algebra*, 5th Edition, Wellesley-Cambridge Press.
3. Sheldon Ross (2014). *A First Course in Probability*, 9th Edition, Pearson.
4. C. L. Liu (2000). *Elements of Discrete Mathematics*, McGraw-Hill.
5. Wayne L. Winston (2004). *Operations Research: Applications and Algorithms*, Duxbury Press.
6. Kanti Swarup, Gupta & Man Mohan (2006). *Operations Research*, Sultan Chand & Sons.

Department	Information Technology
Course Title	Advanced Analysis of Algorithms
Course Code	CS-601
Pre-requisite	None
Credit Hours	3
Objectives	• Provide comprehensive knowledge of modern computer algorithms and solving scientific and engineering problems efficiently and accurately. • The students will be guided how to analyze complex algorithms comparing efficiencies of these algorithms. • Students will not only be taught the design of the existing algorithms but on the other hand it will be focused to teach them designing techniques using rigorous mathematical approaches. • The students will be motivated to think about procedures solving real world problems optimally and correctly. • Real world problem will be taken as examples to create feelings about the usefulness of this course

Learning

Outcomes:

Upon successful completion of this course students will be able to:

- Argue and prove the correctness of algorithms using rigorous mathematical techniques taught in this course.
- Analyze average and worst-case running times of given algorithm.
- Describe the divide-and-conquer technique and arguing when an algorithmic design calls this approach. Derive and solve recurrence relations describing performance of divide-and-conquer algorithms.
- Describe advanced topics such as dynamic programming and greedy approach and reason to use these approaches for a particular situation.
- Integrate dynamic programming and recursive approach improving efficiency of an algorithm.
- Employing graphs to model science and engineering problems, and to reason about when it is appropriate to use it optimally.
- Analyze and design algorithms on further advanced topics such as computational geometry, operations research, cryptography, number theoretic, algorithms etc.

Course Outline

Unit 1	Algorithm analysis techniques: worst-case and amortized, average-case, randomized, competitive, Asymptotic notations, approximation.
Unit 2	Basic algorithm design techniques: greedy, iterative, incremental, divide-and-conquer, dynamic programming, randomization.

Unit 3	Search techniques, Divide-and conquer technique, Randomized algorithms, Heuristic algorithms.
Unit 4	Brute Force approach, Backtracking, String matching, pattern matching.
Unit 5	NP completeness and NP completeness proofs.

Recommended books/reference material

- Introduction to Algorithms, 4th Edition 2022, by T. H. Cormen.
- Algorithm Design, 2012, by Jon Kleinberg and Eva Tardos (Reference Book).
- P, Krishna Sankar & N P, Shangaranarayane. (2018). Design and Analysis of Algorithms.

Department	Information Technology
Course Title	Advanced Operating Systems
Course Code	CS-602
Pre-requisite	None
Credit Hours	3
Objectives	The main purpose of the course is to distinguish distributed Operating Systems from uniprocessor OSes mainly based on the topics - Communication, Synchronization, Processes, File Systems and Shared Memory. Also understand the purpose, structure and functions of distributed operating systems.
Learning Outcomes:	Upon successful completion of this course students will be able to: • Describe the general architecture of distributed concepts. • Master understanding of design issues associated with modern operating systems. • Understand the system resources sharing and load distribution among the systems and users. • Be familiar with protection and security mechanisms.

Course Outline

Unit 1	OVERVIEW. Multi-Threading and Introduction to Distributed Operating Systems. DISTRIBUTED SYSTEMS. Parameters of Network topology, Communication structure, Contention, Robustness and Design Issues.
---------------	--

Unit 2	DISTRIBUTED FILE SYSTEM DFS Structure, Naming structures, Caching and Consistency, Stateless and Stateful file Services, Andrew file system.
Unit 3	DISTRIBUTED COORDINATION Events, Distributed Mutual Exclusion, Atomicity, Concurrency Control, Time-stamping, Election Algorithms, Reaching Algorithms
Unit 4	DISTRIBUTED DEADLOCK DETECTION Deadlock Prevention and Schemes, Central and Distributed approach to deadlock Detection.
Unit 5	REAL-TIME SYSTEMS Real-time Kernels and Scheduling

Recommended books/reference material

- Operating System Concepts, 10th Edition ,By Abraham Silberschatz, Peter B. Galvin, Greg Gagne 2021
- Distributed Operating Systems, Principles and Paradigms, 2/E By Andrew Tanenbaum 2006

Department	Information Technology
Course Title	Advanced Computer Architecture
Course Code	CS-603
Pre-requisite	None
Credit Hours	3

Objectives	• This course is aimed at the hardware aspects of parallel computer architectures including the design and protocols evaluation for memory coherence, inter-connection networks and system scalability. • Advanced topics in this course will cover multiprocessors on a chip, reconfigurable computing and power aware designs. • Various coarse-grained and fine-grained architectures with reference to SIMD and MIMD designs will also be studied.
-------------------	--

Learning Outcomes:	Upon successful completion of this course students will be able to: • Know the classes of computers, and new trends and developments in computer architecture • Understand pipelining, instruction set architectures • Understand the various techniques to enhance a processors ability to exploit Instruction-level parallelism (ILP), and its challenges. • Understand exploiting ILP using dynamic scheduling, multiple issue. • Understand multithreading by using ILP and supporting thread-level parallelism (TLP).
---------------------------	---

- Understand the performance and efficiency in advanced multiple-issue processors.
- Understand VLIW architectures, their performance and case studies.
- Understand the structure and performance of SIMD and MIMD Architectures.

Course Outline

Unit 1	Quantitative Principles of Computer Design
	The Task of a Computer Designer
	Technology and Computer Usage Trends
	Cost and Trends in Cost
	Measuring and Reporting Performance
	Benchmarks and metrics
Unit 2	Instruction Set Principles and Examples
	Classification of Instruction Set Architectures
	Instruction Formats and Semantics
	Memory Addressing Modes
	Operations in the Instruction Set
	Encoding and Instruction Set
	The Role of Compilers
Unit 3	Advanced Pipelining and Instruction-Level Parallelism
	Basic Pipeline Operations
	Data and Control Pipeline Hazards
	Instruction-Level Parallelism
	Dynamic Instruction Scheduling and Branch Prediction
Unit 4	Thread Level Parallelism
	Multiprocessor Systems and Applications
	Centralized Shared-Memory Architectures
	Distributed Shared-Memory Architectures
	Execution Synchronization
	Models of Memory Consistency
Unit 5	Data Level Parallelism (Time Permitting)
	Vector Processing
	Support for Multimedia Applications
	Graphics Processing Units

Recommended books/reference material

- Computer Architecture: A Quantitative Approach, 5th Edition John L. Hennessy And David A. Patterson. Morgan Kaufmann Publishers 2017
- James, I.Samuel Peter & Angeline, D.Magdalene Delighta. (2018). Advanced Computer Architecture.
- Advanced Computer Architecture: 13th Conference, ACA 2020, Kunming, China, August 13–15, 2020, Proceedings (Communications in Computer and Information Science Book 1256)
- Part of: Communications in Computer and Information Science (384 books) by Dezun Dong, Xiaoli Gong, et al. Sep 4, 2020

Department Information Technology

Course Title Theory of Computation

Course Code CS-604

Pre-requisite None

Credit Hours 3

Objectives

- Define and describe formal models of computation, such as finite automata, pushdown automata, and Turing machines;
- Give examples of languages and computational problems appropriate for different models of computation;
- Create proofs for statements regarding formal models of computation;
- Describe class-based resource usage models, including time and space complexity;
- Apply NP-completeness concepts to create proofs regarding the computational complexity of novel problems;
- Use basic concepts and explain implications of modern complexity theoretic approaches to advanced topics such as randomization, proof complexity, and quantum computing.

Learning Outcomes:

Upon successful completion of this course students will be able to

- Discuss key notions of computation, such as algorithm, computability, decidability, reducibility, and complexity, through problem solving.
- Explain the models of computation, including formal languages, grammar and automata, and their connections.
- State and explain the Church-Turing thesis and its significance.
- Analyze and design finite automata, pushdown automata, Turing machines, formal languages, and grammars.
- Solve computational problems regarding their computability and complexity and prove the basic results of the theory of computation.

Course Outline

Unit 1	Mathematical Tools and Techniques: Logic and Proofs, Sets, Functions and Equivalence Relations, Languages. Proofs, Principle of Mathematical Induction, Recursive Definitions, Structural Induction.
Unit 2	Regular Languages and Regular Expression. Nondeterministic Finite Automata with A-Transitions, Kleene's Theorem. Regular, Non-regular Languages, Criteria for Regularity, Minimal Finite Automata
Unit 3	Context-Free Grammars, Regular Grammars, Derivation Tree and Ambiguity. Context-Free, Non-Context-Free Languages. Pushdown Automata: Definitions and Examples, Deterministic Pushdown Automata
Unit 4	Turing Machines: Definitions and Examples, Non-deterministic TM, Universal TM Recursively Enumerable and Languages: Recursively Enumerable and Recursive, Enumerating a Language, More General Grammars, Context-Sensitive Languages and the Chomsky Hierarchy
Unit 5	Unsolvable Problems: A No-recursive Language and an Unsolvable Problem, The Halting Problem, Unsolvable Problems Involving TMs, Rice's Theorem, Post's Correspondence Problem Computable Functions: Primitive Recursive Functions, PRF and Bounded Operations, Unbounded. Minimalization and μ -Recursive Functions, Godel Numbering

Recommended books/reference material

- John Martin. (2010). Introduction to languages and the theory of computation, (4th ed.). New York: McGraw-Hill Science/Engineering/Math.
- Turlakakis, George J. (2012). Theory of computation. Hoboken: Wiley.
- Michael Sipser, Introduction to the Theory of Computation. 2nd edition

Department	Information Technology
Course Title	Theory of Programming Languages
Course Code	CS-605
Pre-requisite	None
Credit Hours	3
Objectives	This course provides an in-depth analysis of the advantages and disadvantages of various programming languages. This study provides the framework for modern compiler design and regular languages. Included is a study of Backus-Naur forms. Lisp and Scheme are explored in further detail.
Learning Outcomes	1. The better understating the underlying theory of programming languages

2. Enable a student to choose the appropriate Language for a Project
3. Learning of formal semantics design for a programming Languages

Course Outline

Unit 1	Introduction: Models of Computation, Syntax and Semantics, Pragmatics, Language Design Principles.
Unit 2	Syntax and Semantics: Context-Free Grammars, Regular Expressions, Attribute Grammars and Static Semantics, Algebraic Semantics, Axiomatic Semantics, Denotational Semantics.
Unit 3	BNF grammars and Syntax, Operational Equivalence, Abstraction and Generalization, Expressions, Assignment Statement, and Control Structures,
Unit 4	Functional Programming: The Lambda Calculus, Operational Semantics, Reduction Order,
Unit 5	Recursive Functions, Logic Programming, Inference Engine, Concurrency

Recommended books/reference material

1. Concepts of Programming Languages, Robert W. Sebesta, 10th edition, 2012
2. Scott, Michael L., Programming Language Pragmatics, 2nd edition, 2006
3. Theory Introduction to Programming Languages, by Anthony A. Aaby, 2004
4. Principles of Programming Languages by Mike Grant Zachary Palmer Scott Smith, John Hopkins University 2016.

Department	Information Technology
Course Title	Advanced Requirement Engineering
Course Code	CS-611
Pre-requisite	None
Credit Hours	3(3-0)
Objectives	The aim of course is enable students in terms of requirement and its types understanding, requirement elicitation, requirement analysis and negotiation, requirement validation and management.
Learning Outcomes	• Understand the basics of requirements engineering. • Understand the social, cultural, political, and technical issues in requirements engineering.

- Understand activities in requirements engineering (requirements elicitation, analysis and negotiation, validation, and management).
- Understand requirements traceability.
- Understand requirements documentation.
- Understand use-case modeling, object-oriented static modeling, functional modeling, and dynamic modeling techniques applicable in requirements engineering.
- Understanding the need for research in selected topics in requirements engineering.
- To know how to apply the learnt concepts, knowledge and techniques to solve real world problems

Course Outline

Unit 1

Introduction to the course, discussion on preliminary concepts of Software Engineering and the importance of Requirements Engineering process.

Fundamental of Software requirements engineering, What are requirements, Some interpretations of "requirement", Types of requirements

Working with the three levels, Product vs. project requirements, Requirements development and management, Reason behind bad requirements, Benefits from a high-quality requirements process.

Requirements development process framework. Good practices of Requirements elicitation, analysis, specification, validation and management.

What are business requirements, Identifying business requirements, vision and scope statement and document, controlling and managing project scope?

Defining requirement elicitation process, requirement elicitation inputs and outputs, requirement elicitation task. What are stakeholders? Importance of stakeholder in requirement elicitation process.

Unit 2

Requirement elicitation techniques, Interviews, Workshops, Focus groups, Observations, Questionnaires, System interface analysis, User interface analysis, Document analysis.

Class activity on requirement elicitation process, Presenting software project idea and class exercise of performing requirement elicitation activities.

Following up after elicitation, Organizing and sharing the notes, documenting open issues, classifying customer input, Some cautions about elicitation, Assumed and implied requirements, Finding missing requirements.

Understanding user requirement, Use cases, The use case approach, Use cases and usage scenarios, Identifying use cases. Introduction to the course, discussion on preliminary concepts of Software Engineering and the importance of Requirements Engineering process.

	Fundamental of Software requirements engineering, What are requirements, Some interpretations of" requirement", Types of requirements
Unit 3	Working with the three levels, Product vs. project requirements, Requirements development and management, Reason behind bad requirements, Benefits from a high-quality requirements process. Requirements development process framework. Good practices of Requirements elicitation, analysis, specification, validation and management. Defining Business Rules, Classification of Business Rules, Documenting Business Rules, Discovering business rules, Business rules and requirements Writing functional requirements, deriving functional requirements from the use case. Characteristics of excellent requirements, Characteristics of requirement statements, Characteristics of requirements collections, Guidelines for writing requirements, System or user perspective
Unit 4	Writing style, Level of detail, Representation techniques, avoiding ambiguity, Avoiding incompleteness, Sample requirements. What are Software Quality attributes or non-functional requirements, classification of non-functional requirements Exploring and specifying non-functional requirements Conflicts related to non-functional requirements, trade off to resolve conflicts, class exercise to elaborates non-functional requirements Requirements modeling, Data flow diagram, Swim lane diagram, State-transition diagram, state table Dialog map, Decision tables, decision trees, Event-response tables
Unit 5	Documenting the requirements, Software requirement specifications, Discussing software requirements specification template Requirements validation and verification, Reviewing requirements, inspection process. Requirements review challenges. Prototyping requirements, Testing the requirements Validating requirements with acceptance criteria, Acceptance criteria, Acceptance tests. Requirements Management: Planning for requirements management, Requirements change management process. Requirements tracing, Requirement Traceability. Backward and forward tractability, tracking and reporting

Recommended books/reference material

- Software Requirements, Wiegers K. &Beatty J., 3rd Edition (2013), Microsoft Press. Reference Book(s):

- Requirements Engineering and Management for Software Development Projects, Chemuturi M., (2013), Springer New York.
- Visual Models for Software Requirements, Beatty J. & Chen A., (2012), Micosoft Press.
- Requirements Engineering, Hull E., Ken Jackson K. & Dick J. 3rd Edition (2011), Springer-Verlag.
- Software & Systems Requirements Engineering: In Practice, Berenbach B., Paulish D. J., Kazmeier J. & Rudorfer A., (2009) McGraw Hill.

Department Information Technology

Course Title **Advanced Global Software Development**

Course Code CS-612

Pre-requisite None

Credit Hours 3

Objectives This course will evaluate and discuss the issues around advanced global software development and techniques for managing distributed projects. Provide the Understanding of Configuration management systems, release management and task assignments in context of distributed projects. Acquire strategies for effectively dividing tasks among teams, controlling the communication among teams, planning tasks and collaborating on modular project with the help of realistic examples.

Learning Outcomes

1. Understand the principles of software engineering in context of global software development.
2. Evaluate and discuss the issues around global software development and techniques for managing distributed projects.
3. Understand Configuration management systems, release management and task assignments in context of distributed projects.
4. Acquire strategies for effectively dividing tasks among teams, controlling the communication among teams, planning tasks and collaborating on modular projects with the help of realistic examples.

Course Outline

Unit 1 **Introduction to Global Software Development**
Introduction to Global Software Development. Global Teams and Organization. Guideline for making the virtual team. The Geography of Coordination. Dealing with Distance.

Unit 2 Architectures and Coordination:
Performance of Networks. Advanced Routing: Routing architecture, Routing between peers (BGP) , IP switching and Multi-Protocol Label Switching MPLS), MPLS Architecture and related protocols, Traffic Engineering (TE) and TE with MPLS, NAT and Virtual Private Networks (L2, L3, and Hybrid),

Unit 3 **Existing Product Technologies**
Reconfiguration of Existing Product Technologies, Identification of Coordination Requirements. Distributed Development

Unit 4	Environments: Software configuration management, Awareness among Configuration Management. IPv6: Why IPv6, basic protocol, extensions and options, support for QoS, security, etc., neighbor discovery, auto-configuration, routing. Changes to other protocols. Application Programming Interface for IPv6.
Unit 5	Concept of Geographically distributed work groups Challenges of Culture: Managing distances and differences in geographically distributed work groups. The Outsourcing Relationship. Facilitating Cross-site Trust, Cooperation, and Social Capital:

Recommended books/reference material

1. Global Software and IT: A Guide to Distributed Development, Projects”, Christof Ebert, Wiley 2011.
2. Global Software Teams: Collaborating Across Borders and Time Zones”, Erran Carmel. Prentice Hall, 2011

Department	Information Technology
Course Title	Software Quality Engineering
Course Code	CS-613
Pre-requisite	None
Credit Hours	3(3-0)
Objectives	Overview of software metrics; Basics of measurements; Goal-based framework for software measurement; Software measure classification; Empirical investigation, principles and techniques; Formal experiments: Planning, principles, types and selection; Measuring internal product attributes: size and structure; Measuring cost and effort; Measuring external product attributes: quality and reliability; Software test metrics; Object-oriented metrics.
Learning Outcomes	Understanding of following concepts • Cost of Software Quality • Engineering Process Areas • Project Management Process Area • Quality Assurance and Defect • Software Quality Assurance • Software Quality Engineering Discipline

Course Outline

Unit 1	Introduction to Software Quality. What is quality? Why software quality? Internal and external quality, Quality Dimensions, quality frameworks and iso-9126
---------------	---

	Software Defects, Errors, Fault, failure, defects, Defect related concepts and relations, types and severity levels of defects Reasons of Poor Quality, Organization Perspective, Buyers Perspective, Development Team perspective Quality Laggards: Characteristics of quality laggards (SQA groups, Quality measurement programs, inspections, defect removal efficiency) Cost of SQA: Conformance and non-conformance cost Prevention cost, appraisal cost, Internal cost external cost, Hidden cost Economics of SQA: Value of software quality, Economic value of software and quality, Economic impact of low and high quality software
Unit 2	Importance of quality Software Requirements, Impact of software requirement quality on project cost and organizational profit Software requirement and quality assurance, Relationship of software requirement and Software Quality assurance Document Requirements Defects, type of requirement, defects, Examples of requirement defects Quality characteristics of Requirements document, Characteristics of Quality Requirement Statements How to write a good quality requirement, Guidelines for Writing Quality Software Requirements.
Unit 3	Quality evaluation of software requirement document. Software requirement specification standards, Practice for Software Requirements Specifications What are software measurements? Significance of software quality measurement. Introduction to software quality measurement Goals of Software Measurement, Data Requirements and Measurement What are software metrics? Why Software metrics? Measurement, Measures, Metrics, Direct and Indirect Measures, Metric classification
Unit 4	Project management metrics, Software development process environmental metrics, product metrics, activity metrics Product based metrics, Process based metrics Models for Quality Assessment, Classification of quality assessment models Generalized Models, Product-Specific Models, overall model, segmented model Model Comparison and Interconnections, Data Requirements and Measurement Selecting Measurements and Models, Selection guidelines and a recommended procedure, steps to select appropriate models and measurements. Basic Ideas and Concepts, Traditional Statistical Analysis Techniques, Correlation Analysis, Linear Regression Models, non-linear regression models, logistic regression Models New Techniques for Risk Identification, Principal component and discriminant analyses, Artificial neural networks and

learning algorithms

Unit 5

Data partitions and tree-based modeling, Pattern matching, and optimal set reduction, Comparisons and Integration
Basic Concepts and General Approaches, Erne domain approach, Input domain approach, Large Software Systems and Reliability Analyses, Characterizing a typical system testing environment.
Reliability Snapshots Using IDRM, commonly used IDRM, Longer-Term Reliability Analyses Using SRGM, Constructing and using TBRM, TBRM Applications
TBRM's impacts on reliability improvement, Implementation and Software Tool Support,
Progress tracking, visualizing reliability growth, Presenting modeling result

Recommended books/reference material

- Software Quality Engineering Testing, Quality Assurance, and Quantifiable Improvement, Jeff Tian, Wiley, 2005.
- Mastering Software Quality Assurance: Best Practices, Tools and Techniques for Software Developers Chemuturi, M., J. Ross Publishing, 2010.
- Software Testing: A Craftsmans' Approach, Jorgensen P.C., CRC Press, 2014.
- Software Quality: Theory and Management, Gillies, A., lulu.com, 2011.
- The art of Software Testing, Myers J. Badgett T. & Sandler C. 3rd Edition, John Wiley & Sons, 2012
- Software Quality Engineering: A Practitioner's Approach, Suryn, W., John Wiley & Sons, NJ, USA, 2014.

Software Engineering, A Practitioner's Approach, Pressman R. S. Palgrave Macmillan, 2005
Software Quality Engineering Testing, Quality Assurance, and Quantifiable

Department	Information Technology
Course Title	Advanced Software Project Management
Course Code	CS-614
Pre-requisite	
Credit Hours	3(3-0)
Objectives	• Differentiate between the skills and roles of functional and technical managers for software efforts and their relationship with other organizations. • Produce specific sections of the plan used to manage the software development and maintenance efforts.

**Learning
outcomes**

- Evaluate software project management practices within an organization and recommend practical improvements based upon your evaluation.
- Apply schedule and cost techniques to determine a Basis of Estimate.
- Identifying the essential software development models, problems areas, domain requirements, design, and quality w.r.t management fundamentals concepts
- Identifying different software processes, initiating the processes and applying planning process task
- Identifying project planning objectives, tools and techniques for SDP, and elements of SDP
- Discussion of life cycle models for an organization, traditional structures of business organization, and organizational planning
- Discussing Estimation concepts, decomposition techniques, and tools
- Using WBS as a Project Management tool with sample and identifying its major steps
- Analysing the core concepts Scheduling and its usage of its tools
- Discussing fundamental concepts of Risk and Change Management
- Revising the basic concepts of Quality

Course Outline

Unit 1	Software Project Management. Core function of management. Knowledge areas of SPM. Integration, Scope, Time. Cost management, HR, Quality. Communication, Risk, Procurement.
Unit 2	Project management process. Charter, Stakeholder identification. Onion Model, Defining objectives and processes. Risk management, Statement of work. Project plan, Planning Space. Rolling wave approach, Tailoring a project plan.
Unit 3	WBS, Types of WBS. Types of Risk and Risk management in SPM. Software Pricing, project scheduling. Estimation Techniques. Project Estimation. Function point analysis.
Unit 4	Use case point, Steps to generate Estimate. Delphi and Wideband Delphi. Out sourcing technique. Software product line. Product line Engineering Framework. Feature Model, Model driven software product line.

Unit 5	Crowd sourcing and types of crowd sourcing. Process of crowd source development. Decision making in project management, Decision analysis process. Formal technical reviews, inspection, work through review.
---------------	--

Recommended books/reference material

- Information Technology Project Management, Schwalbe, K., 9th Edition (2019), Cengage Learning.
- Software Project Management, Hughes, B. & Cotterell, M., 6th Edition (Oct. 2017). McGraw-Hill Higher Education.
- Effective Project Management, Robert, K., Wiley, 2011.

Department	Information Technology
Course Title	Image Processing
Course Code	CS-615
Pre-requisite	None
Credit Hours	3
Objectives	1. To provide knowledge on basic concepts of image and its processing techniques 2. To provide knowledge on Enhancement, Restoration, Segmentation techniques 3. To provide hand on experience of image processing techniques using MATLAB
Learning Outcomes:	1. Students will gain knowledge on fundamental concepts of a digital image processing System. 2. Students will develop skill of developing new algorithms in signal and image processing Applications. 3. Students will develop skill on MATLAB implementation of different signal and image processing techniques

Course Outline

Unit 1	Introduction: Elements of digital image processing, Image model, Sampling and quantization, Relationships between pixels, Image Enhancement: Enhancement by point processing, Spatial filtering, Enhancement in the frequency domain, Colour Image Processing
Unit 2	Image Segmentation: Discontinuity detection, Edge linking and boundary detection, Tresh holding, Region oriented segmentation, Use of motion for segmentation
Unit 3	Introduction to image registration, Techniques of image registration, Representation and Description: Boundary description, Regional description

Unit 4	Morphological Image Processing: Dilation and Erosion, Opening and Closing, Some basic morphological algorithms, Extensions to gray level images
Unit 5	Image Transforms: Discrete Fourier Transform, Discrete Cosine Transform, Haar Transform, Hadamard Transform

Recommended books/reference material

1. Digital Image Processing, R. C. Gonzalez & R. E. Woods, 3rd edition, Prentice Hall, 2008, ISBN 9780131687288.
2. Digital Image Processing (3rd Edition) by Rafael C. Gonzalez, Prentice Hall; 3rd edition (2007)
3. Understanding Digital Signal Processing (3rd Edition) by Richard G. Lyons, Prentice Hall; 3rd edition (2010)

Department	Information Technology
Course Title	Computer Vision
Course Code	CS-616
Pre-requisite	None
Credit Hours	3
Objectives	1. To develop an understanding of the basic principles and techniques of image understanding, and to develop skills in the design and implementation of computer vision software. 2. To develop a computer-based system with the capabilities of the human eye-brain system. 3. To explore some of the basic principles and techniques which are currently being used in real-world computer vision systems.
Learning Outcomes:	Understand and master basic knowledge, theories and methods in computer vision. To Analyze, evaluate and examine existing practical computer vision systems. Communicate effectively and work in teams to develop a working computer vision system. Critically review and assess scientific literature in the field and apply theoretical knowledge to identify the novelty and practicality of proposed methods.

Course Outline

Unit 1	Concepts behind computer-based recognition and extraction of features from raster images. applications of vision systems and their limitations. . Overview of early, intermediate and high level vision.
Unit 2	Segmentation: region splitting and merging; quadtree structures for segmentation; mean and variance pyramids
Unit 3	Computing the first and second derivatives of images using the Sobel and Laplacian operators. grouping edge points into straight lines by means of the Hough transform; limitations of the Hough transform; parameterization of conic sections.

Unit 4	Perceptual grouping: failure of the Hough transform; perceptual criteria; improved Hough transform with perceptual features; grouping line segments into curves.
Unit 5	Camera models, 3D vision, Triangulation principle, Stereoscopy, Generating 3D images from 2D views .

Recommended books/reference material

1. Computer Vision: A Modern Approach 2nd ed, By David Forsyth, Jean Ponce, Prentice Hall, 2011.
2. Computer Vision, by Linda G. Shapiro, George C. Stockman, Prentice Hall, 2001.
3. Handbook of Mathematical Models in Computer Vision, by Nikos Paragios, Yunmei Chen, Olivier Faugeras, Birkhäuser, 2006.

Department	Information Technology
Course Title	Deep learning in computer vision
Course Code	CS-617
Pre-requisite	None
Credit Hours	3
Objectives	1. Design and implement an adequate data extraction and augmentation strategy 2. Implement Deep Learning algorithms with open source frameworks 3. Evaluate performances of the designed approaches in terms of common quality metrics
<u>Learning Outcomes:</u>	1. Build and train deep neural networks, implement vectorized neural networks and identify architecture parameters 2. Use best practices to train and develop test sets, analyze bias/variance for building Deep Learning applications and apply optimization algorithms. 3. Apply deep learning to image, video, and other 2D/3D data for visual detection and recognition tasks.

Course Outline

Unit 1	Colour systems, Spatial sampling in images and video
Unit 2	Key methods of motion estimation in video Unsupervised learning Supervised learning Artificial Neural Networks, MLP as a supervised learning approach
Unit 3	CNN: Fundamentals (convolution, pooling, normalization, non-linearity layers) First architectures (LeNet, AlexNet).
Unit 4	Training Deep CNNs:

Unit 5	Problem of training data, domain dependent data augmentation, Optimization algorithms, transfer learning, deeper architectures , Information fusion with CNNs Temporal aspects in Deep Learning: RNN, LSTM, 2D+t ConvNets , Applications case studies
---------------	--

Recommended books/reference material

1. Pattern Recognition and Machine Learning, Bishop, C., 2006
2. Deep Learning, A practitioner's approach, Adam Gibson and Josh Patterson, O'Reilly, First Edition, 2017.
3. Deep Learning with Python, Francois Chollet, Manning Publications Co, First Edition, 2018.

Department	Information Technology
Course Title	Advance Artificial Intelligence
Course Code	CS-618
Pre-requisite	None
Credit Hours	3
Objectives	1. Know various AI search algorithms (tree search, uninformed, informed, and heuristic), understand different types of AI agents, know how to build simple knowledge-based systems. 2. Ability to apply knowledge representation, reasoning, and machine learning techniques to real-world problems.
Learning Outcomes:	1. Know various AI search algorithms (tree search, uninformed, informed, and heuristic), understand different types of AI agents, and know how to build simple knowledge-based systems. 2. Ability to apply knowledge representation, reasoning, and machine learning techniques to real-world problems

Course Outline

Unit 1	Introduction to Artificial Intelligence, Foundation, The status of knowledge.
Unit 2	State Space Search, State Spaces, Environments and Problems, Production Systems, State Space Search Control, Game Search Control
Unit 3	Knowledge Representation, Knowledge Representation with Classical Logic, Propositional logic, 1st order logic, Online KBs ... SUMO, WordNet, Cyc, Conversion to CNF (FOF to CNF tables), Inference with resolution
Unit 4	Knowledge Representation with Non-classical Logic, Uncertain epistemic status, Imprecise assertional status, Taxonomies, ontologies, and knowledge bases
Unit 5	Machine Learning, Architecture, Decision Trees, Neural Networks, Genetic Algorithms

Recommended books/reference material

1. Artificial Intelligence: A Modern Approach (Pearson Series in Artificial Intelligence): Pearson Series in Artificial Intelligence (1 books) by Stuart Russell and Peter Norvig | Apr 28, 2020
2. Pro Deep Learning with TensorFlow 2.0: A Mathematical Approach to Advanced Artificial Intelligence in Python by Santanu Pattanayak | Jan 1, 2023

Department	Information Technology
Course Title	Deep Learning
Course Code	CS-619
Pre-requisite	None
Credit Hours	3
Objectives	To make students learn the tools and techniques required in handling large amounts of datasets and to uncover various deep learning methods in NLP, Neural Networks etc.
Learning Outcomes:	1. Design convolution networks for handwriting and object classification from images or video. 2. Design recurrent neural networks with attention mechanisms for natural language classification, generation, and translation.

Course Outline

Unit 1	Introduction to Deep learning, Review of Linear classification (Multi-class Support Vector Machines, Soft max) and Regularization, Gradient Descent & Stochastic Gradient Descent (SGD), Back propagation (Intuitions, back propagation as flow graph)
Unit 2	Introduction to Neural Networks (model of a biological neuron, activation functions, neural net architecture, representational power, etc.), Building Neural Networks (data preprocessing, loss functions, weight initialization, regularization, dropout, batch normalization), Learning Neural Networks (Learning and Evaluation gradient checks, sanity checks)
Unit 3	Variants of SGD (momentum, Adagrad/RMSprop, ADAM), Introduction to Convolutional Neural Networks (CNN) and its components (Convolution and Pooling Layers), Convolutional Neural Network case studies (AlexNet/ZFNet/VGGNet), Understanding and Visualizing Convolutional Neural Networks, Convolutional networks for other visual Recognition Tasks (Localization, Detection, Segmentation, etc.)
Unit 4	Transfer Learning and Fine-tuning Convolutional Neural Networks, Introduction to Natural Language Processing (NLP), Learning word and sentences embedding (wordvec, glove, sentvec), Introduction to recurrent networks (RNNs, LSTMS, etc.), Applications of Recurrent neural networks to different NLP tasks (e.g. sentiment analysis, parsing, NER tagging, etc.)

Unit 5	Introduction to Reinforcement Learning and Q-Learning, Deep Q-Networks (DQN) and Game playing using DQN, Introduction to Policy gradients and their applications.
---------------	---

Recommended books/reference material

1. Neural Networks and Deep Learning: A Textbook by Charu C. Aggarwal | Aug 26, 2018
2. Neural Networks and Deep Learning: Neural Networks & Deep Learning, Deep Learning, Blockchain Blueprint by Pat Nakamoto | Jul 1, 2018

Department	Information Technology
Course Title	Machine Learning
Course Code	CS-621
Pre-requisite	None
Credit Hours	3
Objectives	To train a model on historical, labelled data (i.e., data for which the outcome is known) in order to predict the value of some quantity on the basis of a new data item for which the target value or classification is unknown.
Learning Outcomes:	1. Understand a wide variety of learning algorithms. 2. Understand how to evaluate models generated from data. 3. Apply the algorithms to a real problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models.

Course Outline

Unit 1	Introduction to machine learning and statistical pattern recognition. Supervised learning: Part I (Graphical models (full Bayes, Naïve Bayes), Decision trees for classification & regression for both categorical & numerical data, Ensemble methods, Random forests, Boosting (Adaboost and Xgboost), Stacking
Unit 2	Part II (Four Components of Machine Learning Algorithm (Hypothesis, Loss Functions, Derivatives and Optimization Algorithms), Gradient Descent, Stochastic Gradient Descent
Unit 3	Linear Regression, Nonlinear Regression, Perceptron, Support vector machines, Kernel Methods, Logistic Regression, Softmax, Neural networks); Unsupervised learning: K-means, Density Based Clustering Methods (DBSCAN, etc.)
Unit 4	Gaussian mixture models, EM algorithm, etc.; Reinforcement learning; Tuning model complexity; Bias-Variance Tradeoff;
Unit 5	Grid Search, Random Search; Evaluation Metrics; Reporting predictive performance

Recommended books/reference material

1. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition by Trevor Hastie, Robert Tibshirani, 2017
2. Pattern Recognition and Machine Learning (Information Science and Statistics) by Christopher M. Bishop, 2016 Pattern Recognition & Machine Learning, 1st Edition.
3. Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series by Kevin P. Murphy, 2012
4. Applied Machine Learning and AI for Engineers: Solve Business Problems That Can't Be Solved Algorithmically by Jeff Prosise, 2022

Department	Information Technology
Course Title	Natural Language Processing
Course Code	CS-622
Pre-requisite	None
Credit Hours	3
Objectives	Develop speech-based applications that use speech analysis (phonetics, speech recognition, and synthesis). Analyze the syntax, semantics, and pragmatics of a statement written in a natural language.
Learning Outcomes:	1. Develop speech-based applications that use speech analysis (phonetics, speech recognition, and synthesis). 2. Analyze the syntax, semantics, and pragmatics of a statement written in a natural language. 3. Develop a conversational agent that uses natural language understanding and generation.

Course Outline

Unit 1	Introduction to NLP, linguistics and NLP tasks, Python libraries for NLP, Text preprocessing and N-grams
Unit 2	Softmax / MAXENT (sequence) classifiers, sequence classifiers for POS & NER, Deep learning based word representations & deep networks for NER, Recurrent networks and language modeling
Unit 3	Statistical machine translation, Machine translation: word alignment, parallel corpora, decoding, evaluation
Unit 4	Modern deep learning machine translation systems (phrase-based, syntactic), Syntax and parsing, co-reference resolution, Tree recursive neural networks for POS tagging
Unit 5	Computational semantics, Question answering, Text summarization, Dialogue systems

Recommended books/reference material

1. Speech and Language Processing, 2nd Edition, Daniel Jurafsky and James Martin, 2008
2. Foundations of Statistical Natural Language Processing, 2nd Edition, Chris Manning and Hinrich Schuetze, 1999
3. Neural Network Methods for Natural Language Processing, 1st Edition, Yoav Goldberg 2017

Department	Information Technology
Course Title	Advanced Computer Networks
Course Code	CS-623
Pre-requisite	None
Credit Hours	3
Objectives	To provide students with a comprehensive introduction to fundamental concepts of data communication and computer networks. To understand the basic encoding and modulation techniques. This course will introduce basic network devices and LAN technologies, IP addressing concepts and subnetting skills, and basic Computer Utility Tools
Learning Outcomes:	The students will have a unified view of data communications and computer networks. They layered architecture approach with reference to OSI Model. The students will have the fundamental knowledge of encoding and modulation techniques, error detection, flow and error control techniques and ad hoc networks. The students will have basic know of Network Simulation and Network Analysis Tools.

Course Outline

Unit 1	Network architecture, Networking principles, Network services and Layered architecture, Future Networks. Advanced Technologies.
Unit 2	Performance of Networks. Advanced Routing: Routing architecture, Routing between peers (BGP) , IP switching and Multi-Protocol Label Switching MPLS), MPLS Architecture and related protocols, Traffic Engineering (TE) and TE with MPLS, NAT and Virtual Private Networks (L2, L3, and Hybrid),
Unit 3	CIDR –Introduction, CIDR addressing, CIDR address blocks and Bit masks. Mobile IP-characteristics, Mobile IP operation, Security related issues. Mobility in networks. Voice and Video over IP (RTP, RSVP, QoS).
Unit 4	IPv6: Why IPv6, basic protocol, extensions and options, support for QoS, security, etc., neighbor discovery, auto-configuration, routing. Changes to other protocols. Application Programming Interface for IPv6.
Unit 5	Ad Hoc Networking: An Introduction, A DoD Perspective on Mobile Ad Hoc Networks, DSDV: Routing over a Multi hop Wireless Network of Mobile Computers, Cluster-Based Networks, DSR: The Dynamic Source Routing Protocol for Multi hop Wireless Ad Hoc Networks.

Recommended books/reference material

1. Computer Networks: A Systems Approach by Larry L. Peterson, Bruce S, Morgan Kaufmann; 5th Edition (March 25, 2011). ISBN-10: 0123850592.

2. Internetworking with TCP/IP Vol –I by Douglas E. Comer, Addison-Wesley; 5th Edition (July 10, 2005). ISBN-10: 0131876716
3. Metro Ethernet by Sam Halabi Publisher: Cisco Press ISBN: 158705096X Computer Networks by A. S. Tanenbaum, Prentice Hall; 5th Edition (October 7, 2010). ISBN-10: 0132126958

Department	Information Technology
Course Title	Security Management & Cryptography
Course Code	CS-624
Pre-requisite	None
Credit Hours	3
Objectives	The course will provide the student with an understanding of the principles of Security Management & Cryptography that are commonly used in business. It will introduce the student to commonly used frameworks and methods and explore critically the suitability and appropriateness of these for addressing today's organizational security needs.
Learning Outcomes	The successful participant will have an understanding of the key themes and principles of information security management and be able to apply these principles in designing solutions to managing security risks effectively.

Course Outline

Unit 1	Introduction Fundamental concepts of Security, Types of Attacks, Social Engineering Attacks Classification traits of malwares, Circulation, Infection, Concealment, Payload capabilities, Web Application Attacks: SQL injection, Cross Site Scripting
Unit 2	Security Management & Cryptography. Client Side Attacks: Cookies, Dos, Man in the Middle, Replay, Developing Security Policy, Deploy and manage Security settings., Security Through Design, Security Through Anti Malware, Fundamentals of Cryptography
Unit 3	OSI Reference Model OSI Reference Model, AES. Standard Network Devices. Network Security Hardware, Firewalls, Types of Firewalls, Spam Filters, Virtual Private Networks Intrusion Detection and Prevention Study, DNS
Unit 4	Network Security: Network Address Translation (NAT): Network Access Control (NAC), Network Protocols, TCP/IP. Wireless Network Security: Wireless Network Attacks, Types of Attacks
Unit 5	Mobile Devices Security Mobile Devices Security, Cloud Security Challenges and Solution, IOT security Challenges

Recommended books/reference material

1. Security+ Guide to Network Security Fundamentals by Mark Ciampa, the Edition
Corporate Computer Society by Randall J.Boyle, 3rd Edition.

Department	Information Technology
Course Title	Distributed Systems
Course Code	CS-624
Pre-requisite	
Credit Hours	3.0
Objectives	• The overall aim of this module is to introduce students to the theory, practice, and advanced techniques associated with implementing large-scale distributed computing systems in Service-Oriented Architectures (SOA). •
Learning Outcomes:	1. Students will be able to gain the technical knowledge necessary to analyze the scalability and interoperability problems associated with large-scale heterogeneous distributed systems and abstracting them into service-oriented architectures. 2. Enable students to become accomplished in the design and implementation of enterprise-level computer applications and associated data-storage as services that can be cost-effectively deployed in scalable Cloud infrastructures.

Course Outline

Unit 1	Distributed system Analysis of the constraints and capabilities of distributed, heterogeneous, large scale information-sharing systems. Demonstrate an appreciation of the problems that can arise in the development of large-scale, heterogeneous, distributed applications. TCP/IP in distributed system TCP/IP-based Network Communication in Java Implementation of TCP/IP using Java UDP-based Network Communication in Java A practical example of UDP in Java
Unit 2	Implementation of UDP in Java JAVA Remote Method Invocations (RMI) JAVA Remote Method Invocations (RMI) – Revised Implementation of RMI in Java
Unit 3	Architecture, XML, Web Services Service-Oriented Architecture Extensible Markup Language XML and JAXB Implementing SOA JAX-WS Web Services

Unit 4 SOA Services

Building SOA Services using JAX-WS
Programming Model of SOA at the Server Side
Client Side Programming- Java SE
Representational State Transfer RESTful Web Services
REST and HTTP
REST VS SOAP

Unit 5 Semantic Web

The Semantic Web (Web3.0) & the Linked Data Cloud
Ontology tree & Sentence Structure
The World Wide Web and Semantic Web
Semantic Web Stack (detailed)

Recommended books/reference material

- Web services & SOA: principles and technology - Papazoglou, M., 2012
- Java EE 6 Development With NetBeans 7 - Heffelfinger, David, 2011
- SOA with REST: principles, patterns & constraints for building enterprise solutions with REST - Erl, Thomas, c2013 [recommended reading]
- Middleware and Cloud Computing: Oracle on Amazon Web Services (AWS), Rackspace Cloud and RightScale (Volume 1) - Frank Munz
- A Semantic Web primer - Antoniou, G., 2012 [independent research]vISOM, P.K. & HOLLEY, K. (2012) "Is Your Company Ready for Cloud? Choosing the Best Cloud Adoption Strategy for Your Business", IBM Press/Pearson, London [ISBN 0-13-259984-8]

Department Information Technology

Course Title Research Methods

Course Code CS-608

Pre-requisite None

Credit Hours 3

Objectives

- Read, interpret, and critically evaluate computer science research.
- Identify, explain, and apply the basic concepts of research, such as variables, operationalization, sampling, reliability, and validity.
- Recognize the ethical issues involved in research, and practice ethical research standards.

- Identify and explain the difference between quantitative, qualitative, and mixed methods research and what types of research questions can be answered with each method.
- Use theory and previous research to create research questions and hypotheses and to identify and analyze the appropriate method and variables needed for research questions.
- Construct a coherent research proposal that includes an abstract, introduction, literature review, research questions, ethical considerations, and methodology

**Learning
Outcomes:**

Upon successful completion of this course students will be able to:

- Develop the knowledge and skills related to computer science research methods.
- Learn the ways of Literature Review in the field of computer science.
- Learn the ways to analyze Literature Review.
- Perform critical analysis of previous work
- Develop quantitative and qualitative instruments (building questionnaires, interview guides etc), as well as choose appropriate data analysis techniques in qualitative and quantitative research and conduct such analysis
- To interpret data produced with various support tools.

Course Outline

Unit 1

INTRODUCTION TO RESEARCH:

- Research Methods in Computer Science
- Analytical vs. Empirical Methods
- Surveys & Case Studies
- Controlled Experiments, Ethnography and Action Research
- Quantitative, Qualitative, and Mixed Methods, Choosing research methods, Validity threats.
- An Empirical Research Framework
- Research Problems, Literature Reviews.

Unit 2

- Introduction to Quantitative Research, Study Designs, Controlled Experiments, Elements and Methods. Example Experiments
- Data Collection Techniques, Analysis and Interpretation of Quantitative Data, Descriptive Statistics,
- Sampling, sampling distribution, Parameter Estimation, Statistical Inference, confidence interval and Hypothesis Testing.

Unit 3

- Tests of significance, test of difference of mean and proportions, t-tests, ANOVA, Chi-square Tests, correlation and regression,
- Review Process, Review guidelines, Validity threats, Review decisions.
- Qualitative Methods, Study Designs, Elements, and Methods, The nature and types of qualitative research.

- | | |
|---------------|---|
| Unit 4 | <ul style="list-style-type: none"> • Data collection methods - primary and secondary sources, Types of data analysis methods • Survey Research, Sampling Methods, Survey Study Designs, Case Studies. |
| Unit 5 | <ul style="list-style-type: none"> • Introduction to Mixed Methods Research, Study Designs and Method. • Writing research papers, purpose, nature and evaluation, content and format, Research Presentations • The Art of Scientific and Technical Writing |

Recommended books/reference material

- Research Design. Qualitative, Quantitative, and Mixed Methods Approaches. By John W. Creswell, Fourth Edition. SAGE Publication, 2014
- The Craft of Research, By Wayne C. Booth, Gregory G. Colomb, Joseph M. Williams, Joseph Bizup, William T. FitzGerald, Third Edition, The University of Chicago Press, 2008
- Research Methodology By Panneerselvam R, 2nd Edition, PHI, 2014

Department: Information Technology

Course Title: Advance Ubiquitous Computing

Course Code: IT-701

Credit Hours: 03

Pre-requisite: None (recommended: Computer Networks, Operating Systems, and Mobile Computing)

Course Objectives

- To explore the fundamental principles, architectures, and enabling technologies of **ubiquitous/pervasive computing systems**.
- Equip students with the ability to **design, develop, and evaluate context-aware applications** for smart environments.
- Develop understanding of **context modeling, sensor-based systems, human-computer interaction, mobile and embedded systems**, and **privacy** in pervasive computing.
- Encourage critical thinking to assess the **socio-technical implications** of ubiquitous computing in real-world applications like smart homes, healthcare, retail, and urban systems.
- Provide hands-on experience in building prototypes using **IoT devices, context-aware systems**, and **middleware frameworks**.

Learning Outcomes

Upon successful completion of the course, students will be able to:

1. Explain the architecture and design challenges of ubiquitous computing systems. (C2 Understand)
2. Apply models and frameworks for **context awareness** and **adaptive systems**. (C3 Apply)
3. Design and prototype **smart environment applications** using sensors and embedded systems. (C6 Create)
4. Evaluate privacy, ethical, and security concerns in pervasive computing. (C5 Evaluate)

5. Analyze key protocols, middleware, and system software enabling ubiquitous computing. *(C4 Analyze)*
6. Integrate ubiquitous computing solutions with **cloud, edge, and mobile platforms**. *(C3 Apply)*
7. Conduct research or case studies on **cutting-edge applications** in domains like health, smart cities, and intelligent transportation systems. *(P4 Articulation)*

Course Outline

Unit 1: Introduction to Ubiquitous Computing

- History and vision (Mark Weiser's vision)
- Characteristics and key features of ubiquitous systems
- Applications in healthcare, transportation, smart homes, retail, and education
- Ubiquitous vs. pervasive vs. mobile computing

Unit 2: Context-Awareness and Smart Environments

- Definition of context and context modeling
- Context acquisition, reasoning, and management
- Context-aware systems and adaptive behavior
- Smart spaces and ambient intelligence
- User modeling and personalization

Unit 3: System Architecture and Middleware

- Architectures for ubiquitous systems
- Middleware and platforms for context-aware computing (e.g., Context Toolkit, OpenIoT)
- Service discovery, messaging, and device interoperability
- Edge, fog, and cloud integration

Unit 4: Sensing, Communication, and Device Integration

- Sensor types and their applications (wearable, ambient, mobile)
- Embedded systems and hardware platforms (Arduino, Raspberry Pi, ESP32)
- Communication protocols: Bluetooth, ZigBee, MQTT, CoAP, LoRa
- Energy efficiency and mobility management

Unit 5: Security, Privacy, and Ethical Issues

- Threats and risks in ubiquitous computing
- Data protection and anonymization
- User consent and ethical data collection
- Case studies: surveillance, location tracking, and health monitoring

Unit 6: Emerging Trends and Case Studies

- Ubiquitous AI and machine learning for context prediction
- Brain-computer interfaces (BCI)
- Digital twins and ubiquitous simulation
- Case studies in smart cities, precision agriculture, assisted living
- Research trends and future directions

Recommended Books / Reference Material

1. Krumm, J. (2018). *Ubiquitous Computing Fundamentals* (2nd Edition). CRC Press.
2. Poslad, S. (2011). *Ubiquitous Computing: Smart Devices, Environments and Interactions*. Wiley.
3. Cook, D. J., & Das, S. K. (2012). *Smart Environments: Technology, Protocols and Applications*. Wiley.
4. Want, R., & Schilit, B. (2021). *Ubiquitous Computing* (Foundations and Trends in Human–Computer Interaction).
5. Journal articles from IEEE Pervasive Computing, ACM Ubicomp Conference, and Springer Smart Environments series.

Department: Information Technology

Course Title: Advance Cloud Computing

Course Code: IT-713

Credit Hours: 03

Pre-requisite: Computer Networks, Operating Systems, and Distributed Systems (recommended)

Course Objectives

- Introduce fundamental principles and architectures of **cloud computing**, including virtualization, service models, deployment models, and resource management.
- Provide understanding of **cloud infrastructure**, data storage, and distributed computing technologies used in large-scale enterprise and scientific applications.
- Enable students to evaluate **cloud services**, platforms, and **security frameworks** in real-world contexts.
- Equip students with knowledge to **design scalable, fault-tolerant cloud applications** using industry-standard tools and platforms.
- Stimulate research on emerging topics such as **serverless computing, edge computing, and container orchestration**.

Learning Outcomes

Upon successful completion of the course, students will be able to:

1. Explain the core concepts, architectures, and models in cloud computing. *(C2 Understand)*
2. Analyze different cloud service models (IaaS, PaaS, SaaS) and deployment types (public, private, hybrid). *(C4 Analyze)*
3. Design and evaluate **scalable and elastic cloud-based systems**. *(C6 Create)*
4. Compare and apply **virtualization and containerization** technologies for resource abstraction and optimization. *(C3 Apply)*
5. Address challenges in **cloud security, identity management, and compliance**. *(C5 Evaluate)*
6. Explore the integration of cloud computing with **big data, AI/ML, and IoT**. *(P4 Articulation)*
7. Critically assess future directions in **edge computing, multi-cloud systems, and serverless computing**. *(C6 Create)*

Course Outline

Unit 1: Introduction to Cloud Computing

- History and evolution of cloud computing
- Key characteristics and benefits

- Service models: IaaS, PaaS, SaaS
- Deployment models: Public, Private, Hybrid, Community
- Case studies of popular providers (AWS, Azure, GCP)

Unit 2: Cloud Architecture and Virtualization

- Cloud architecture and components
- Virtualization: hypervisors (Xen, KVM, VMware), containers (Docker, LXC)
- Resource provisioning and elastic scaling
- Infrastructure orchestration: OpenStack, Kubernetes, Terraform (overview)

Unit 3: Cloud Storage and Data Management

- Cloud storage models: block, object, and file storage
- Distributed file systems (HDFS, GFS)
- Data consistency and availability (CAP theorem, eventual consistency)
- NoSQL databases in the cloud (MongoDB, Cassandra)

Unit 4: Cloud Application Development and Platforms

- Platform as a Service (PaaS)
- Cloud-native application design
- Serverless computing and Function-as-a-Service (AWS Lambda, Azure Functions)
- Microservices and 12-factor applications
- DevOps in the cloud (CI/CD pipelines)

Unit 5: Security and Privacy in Cloud Computing

- Threats, vulnerabilities, and risk analysis
- Identity and access management (IAM)
- Data encryption at rest and in transit
- Regulatory compliance (GDPR, HIPAA)
- Multi-tenancy and isolation

Unit 6: Emerging Trends and Case Studies

- Edge and fog computing
- Multi-cloud and hybrid cloud management
- Green cloud computing and sustainability
- AI/ML on the cloud: managed services
- Case studies in healthcare, finance, education, and government
- Research directions in cloud federation and cloud standardization

Recommended Books / Reference Material

1. Rajkumar Buyya, Christian Vecchiola, & S. Thamarai Selvi. *Mastering Cloud Computing* (Morgan Kaufmann, 2013).
2. Thomas Erl. *Cloud Computing: Concepts, Technology & Architecture* (Prentice Hall, 2013).
3. George Reese. *Cloud Application Architectures* (O'Reilly, 2009).
4. Toby Velte, Anthony Velte, & Robert Elsenpeter. *Cloud Computing: A Practical Approach* (McGraw-Hill, 2010).
5. Journal articles from IEEE Cloud Computing, ACM Digital Library, and Springer Cloud Computing journals.
6. Documentation from AWS, Microsoft Azure, and Google Cloud Platform.

Department: Information Technology
Course Title: Advance Wireless Sensor Networks
Course Code: IT-715
Credit Hours: 03
Pre-requisite: Computer Networks, Embedded Systems (recommended)

Course Objectives

- Introduce students to the fundamentals of **wireless sensor networks (WSNs)** including architecture, design, protocols, and real-world applications.
- Enable students to design, develop, and analyze **WSN systems** using simulation tools and analytical methods.
- Cover communication protocols, energy efficiency, and scalability challenges specific to **resource-constrained wireless devices**.
- Provide insight into WSN security, data aggregation, and integration with IoT and cloud platforms.
- Stimulate research thinking on **adaptive, self-organizing, and intelligent sensor networks**.

Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Explain the architecture, operation, and characteristics of wireless sensor networks. *(C2 Understand)*
2. Analyze communication protocols for medium access control, routing, and transport layers in WSNs. *(C4 Analyze)*
3. Model and simulate WSN deployments using appropriate tools. *(P2 Manipulation)*
4. Evaluate WSN performance in terms of latency, power consumption, coverage, and reliability. *(C5 Evaluate)*
5. Design and propose efficient sensor network architectures for specific applications. *(C6 Create)*
6. Understand WSN security concerns and apply basic cryptographic and data protection techniques. *(C3 Apply)*
7. Explore integration of WSNs with **IoT, Edge Computing, and Cloud Services**. *(P4 Articulation)*

Course Outline

Unit 1: Introduction to Wireless Sensor Networks

- Overview and applications of WSNs
- Comparison with traditional networks
- Challenges in WSNs: energy, memory, computation, latency
- Sensor node architecture and components
- Network architecture and design goals

Unit 2: Medium Access Control (MAC) Protocols

- MAC challenges in WSNs
- Energy-efficient MAC protocols
- S-MAC, T-MAC, B-MAC
- Adaptive duty-cycling techniques
- TDMA and contention-based access schemes

Unit 3: Routing Protocols

- Routing challenges and metrics in WSNs
- Flat routing: Directed Diffusion, Rumor Routing
- Hierarchical routing: LEACH, TEEN, PEGASIS
- Location-based routing: GEAR, GAF
- Data-centric vs. address-centric routing

Unit 4: Localization, Time Synchronization & Coverage

- Node localization techniques (GPS-free)
- Time synchronization protocols: TPSN, RBS
- Coverage models and connectivity
- Mobility in WSNs
- Sensor deployment strategies

Unit 5: Power Management and Energy Harvesting

- Energy-efficient communication
- Sleep/wake scheduling
- Energy-aware routing and MAC
- Energy harvesting techniques
- Battery models and lifetime prediction

Unit 6: Security and Fault Tolerance in WSNs

- Threats and attacks on WSNs
- Lightweight cryptographic techniques
- Key management in resource-constrained environments
- Secure routing and data aggregation
- Fault detection and recovery strategies

Unit 7: WSN Platforms, Programming & Simulation

- Sensor operating systems: TinyOS, Contiki
- WSN programming models
- Simulation tools: NS-3, Cooja, TOSSIM
- Performance evaluation metrics
- Case studies on environmental monitoring, smart agriculture, military systems

Unit 8: Advanced Topics and Research Trends

- Wireless Body Area Networks (WBANs)
- Integration with Internet of Things (IoT)
- Cognitive WSNs
- Mobile sink strategies and delay-tolerant WSNs
- WSN in smart cities and edge computing environments

Recommended Books / Reference Material

1. Holger Karl and Andreas Willig, *Protocols and Architectures for Wireless Sensor Networks*, Wiley, 2005.
2. Kazem Sohraby, Daniel Minoli, and Taieb Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*, Wiley, 2007.
3. Bhaskar Krishnamachari, *Networking Wireless Sensors*, Cambridge University Press, 2005.

4. C. Siva Ram Murthy and B. S. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*, Pearson, 2004.
5. Research papers from IEEE Sensors Journal, ACM Transactions on Sensor Networks, and Ad Hoc Networks (Elsevier).
6. Online documentation of TinyOS, Contiki, and NS-3.

Department: Information Technology

Course Title: Advance Internet of Things

Course Code: IT-715

Credit Hours: 03

Pre-requisite: Computer Networks, Embedded Systems (recommended)

Course Objectives

- Introduce the foundational concepts of the Internet of Things (IoT), including architecture, devices, communication protocols, and data handling.
- Enable students to design, implement, and evaluate IoT systems using sensors, actuators, and microcontrollers.
- Cover end-to-end IoT solutions involving data collection, cloud integration, analytics, and real-time decision-making.
- Explore industry use-cases and research challenges in IoT across domains like smart homes, healthcare, agriculture, and industry.
- Equip students with the tools and techniques required to prototype IoT systems and integrate them with modern cloud and edge platforms.

Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Explain the architecture and components of IoT systems. (*C2 Understand*)
2. Identify and evaluate appropriate communication technologies and protocols for IoT. (*C3 Apply*)
3. Analyze and implement data acquisition and transmission in resource-constrained environments. (*P2 Manipulation*)
4. Develop and simulate IoT applications using microcontrollers and software frameworks. (*C6 Create*)
5. Design secure and scalable IoT systems integrated with edge/cloud computing platforms. (*P4 Articulation*)
6. Evaluate societal, ethical, and environmental implications of IoT deployments. (*C5 Evaluate*)

Course Outline

Unit 1: Introduction to IoT

- Definition and scope
- History, evolution, and trends
- Characteristics of IoT systems
- IoT reference architecture
- IoT vs. traditional embedded systems vs. cyber-physical systems

Unit 2: IoT Hardware and Device Platforms

- Microcontrollers (Arduino, ESP32, Raspberry Pi)
- Sensors and actuators: interfacing and control

- Device communication and power management
- GPIO, I2C, SPI, UART interfaces

Unit 3: Communication and Networking in IoT

- Network layer protocols: IPv6, 6LoWPAN
- Data link protocols: ZigBee, BLE, LoRaWAN, NB-IoT
- Transport protocols: MQTT, CoAP, AMQP
- Edge computing and fog computing

Unit 4: Data Acquisition and Processing

- Analog to digital conversion
- Signal processing for sensor data
- Real-time data transmission
- Local vs. remote processing (edge vs. cloud)

Unit 5: IoT Platforms and Cloud Integration

- Overview of cloud services for IoT (AWS IoT, Google IoT Core, Azure IoT Hub)
- Data ingestion pipelines
- Storage, visualization, and dashboard tools
- RESTful APIs and webhooks

Unit 6: Security and Privacy in IoT

- Vulnerabilities in IoT devices
- Data encryption and secure communication
- Identity and access management
- Blockchain in IoT
- Case studies: DDoS via IoT, firmware attacks

Unit 7: Industrial and Research Applications of IoT

- Smart homes and buildings
- Smart cities and transportation
- Industrial IoT (IIoT) and Industry 4.0
- Healthcare and wearable technology
- Precision agriculture

Unit 8: Recent Trends and Challenges

- AIoT – Artificial Intelligence in IoT
- Interoperability and standardization
- Energy-efficient networking
- Legal, ethical, and environmental considerations
- Future research directions in IoT

Recommended Books / Reference Material

1. Pethuru Raj, Anupama C. Raman. *The Internet of Things: Enabling Technologies, Platforms, and Use Cases*, CRC Press, 2017.
2. Arshdeep Bahga, Vijay Madisetti. *Internet of Things: A Hands-On Approach*, VPT, 2014.
3. Adrian McEwen and Hakim Cassimally. *Designing the Internet of Things*, Wiley, 2013.

4. Dieter Uckelmann et al. *Architecting the Internet of Things*, Springer, 2011.
5. Jan Holler et al. *From Machine-to-Machine to the Internet of Things*, Elsevier, 2014.
6. Research papers from IEEE IoT Journal, ACM Transactions on IoT, and relevant conference proceedings.
7. Online documentation for platforms like AWS IoT Core, Google Cloud IoT, Node-RED, and Arduino IDE.

Department: Information Technology

Course Title: Introduction to Software Defined Networks

Course Code: IT-717

Credit Hours: 03

Pre-requisite: Computer Networks

Course Objectives

- To introduce the motivation, principles, and architecture of Software Defined Networking (SDN).
- To understand the separation of control and data planes and how SDN enables network programmability.
- To familiarize students with SDN controllers, southbound and northbound APIs, and OpenFlow protocol.
- To explore SDN use-cases in data centers, cloud computing, and WANs.
- To analyze security, scalability, and performance issues in SDN-based architectures.
- To develop the ability to simulate and experiment with SDN tools and platforms.

Learning Outcomes

After completing this course, students will be able to:

1. Explain the need for and advantages of Software Defined Networking. *(C2 Understand)*
2. Differentiate between traditional and SDN architectures. *(C3 Apply)*
3. Design and configure SDN networks using appropriate protocols and platforms. *(P2 Manipulation)*
4. Use SDN controllers and APIs to program network behavior. *(C4 Analyze)*
5. Evaluate and apply SDN solutions to real-world networking problems. *(C5 Evaluate)*
6. Investigate security, scalability, and fault tolerance in SDN environments. *(C6 Create)*

Course Outline

Unit 1: Introduction to SDN

- Traditional networking vs. SDN
- Control plane vs. data plane
- Benefits and challenges of SDN
- SDN architecture and layers
- Centralized vs. distributed control

Unit 2: OpenFlow and Southbound Interfaces

- Introduction to OpenFlow
- Flow tables and match-action rules
- OpenFlow protocol versions
- Packet processing and forwarding
- Switch-controller communication

Unit 3: SDN Controllers

- Role of controllers in SDN
- Centralized and distributed controllers
- Popular SDN controllers: Ryu, ONOS, OpenDaylight, Floodlight
- Controller-to-device (southbound) and controller-to-app (northbound) APIs
- Controller clustering and scalability

Unit 4: Northbound APIs and Network Programming

- Need for northbound APIs
- RESTful APIs and SDKs
- Network programmability with Python/Java
- Policy definition and enforcement
- Intent-based networking

Unit 5: SDN Applications and Use Cases

- Traffic engineering and load balancing
- Network function virtualization (NFV)
- Security: firewalls, DDoS detection
- Data center networking
- SDN in ISPs, cloud providers, and enterprise networks

Unit 6: SDN Virtualization and Emulation

- Network slicing and virtualization
- Overlay networks and tunneling (VXLAN, GRE)
- Tools for SDN simulation and emulation: Mininet, GNS3, NS-3
- SDN in containerized and virtualized environments (Docker, Kubernetes)

Unit 7: Security and Fault Tolerance in SDN

- Threats and attack surfaces in SDN
- Secure controller-switch communication
- Authentication and authorization in SDN
- Controller failure and recovery
- Trust and verification frameworks

Unit 8: Research Trends and Future of SDN

- SDN and 5G
- SD-WAN
- AI/ML for SDN
- Open standards and open-source SDN
- Future challenges and emerging trends

Recommended Books / Reference Material

1. Kreutz, D., Ramos, F. M. V., Verissimo, P., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). *Software-Defined Networking: A Comprehensive Survey*. IEEE Communications Surveys & Tutorials.

2. Feamster, N., Rexford, J., & Zegura, E. (2014). *The Road to SDN: An Intellectual History of Programmable Networks*. ACM SIGCOMM.
3. Thomas D. Nadeau, Ken Gray. *SDN: Software Defined Networks*. O'Reilly Media, 2013.
4. Paul Goransson, Chuck Black, Timothy Culver. *Software Defined Networks: A Comprehensive Approach*, 2nd Edition, Morgan Kaufmann, 2016.
5. Mininet documentation and OpenFlow specifications (<https://www.opennetworking.org/>).
6. Research papers from IEEE, ACM, and Elsevier on SDN technologies and use cases.