

Curriculum Master of Science in Computer Science

(2026)



Rawalpindi Women University, Rawalpindi

Table of Contents

1. Curriculum for Master of Science in Computer Science (MS (CS))	3
1.1 Program Vision	3
1.2 Program Mission.....	3
1.3 Eligibility Criteria	3
1.4 Degree Duration.....	3
1.5 Degree Completion Requirements	4
1.6 Program Objectives and Outcomes.....	4
2. Area Covered in MS (CS) Program	5
3. Scheme of Study of MS Computer Science in the light of HEC Curriculum	5
4. Roadmap of MS (CS) Program.....	7
5. Core Courses	8
6. Course Outline Core Courses	9
7. Elective Courses	16
8. Course Outline Elective Courses	17
9. Mandatory Courses	53
10. Course Outline Mandatory Courses	54

1. Curriculum for Master of Science in Computer Science (MS (CS))

1.1 Program Vision

To be a leading center of excellence in computer science, empowering women to drive innovation in Artificially Intelligent era fostering emerging technologies. Through interdisciplinary research and critical thinking, we aim to create a secure and intelligent digital future for a better tomorrow

1.2 Program Mission

To empower women in computer science through excellence in research, innovation, and critical thinking. We are committed to advancing emerging technologies and AI to create a secure, intelligent digital future. Through collaboration and learning, we inspire and equip future technology leaders

1.3 Eligibility Criteria

The minimum requirements for admission in a MS (CS) 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"

The following core courses are recommended to be completed before entering the MS (CS) program.

1. Analysis of Algorithms
2. Assembly Lang. / Computer Architecture
3. Computer Networks
4. Computer Programming
5. Data Structures
6. Database Systems
7. Operating Systems
8. Software Engineering
9. Theory of Automata

A student selected for admission having deficiency in the above stated courses may be required to study a maximum of FOUR courses offered according to RWU policy, which must be passed in the first two semesters. Deficiency courses shall be determined by the Graduate Studies Committee, before admitting the student.

1.4 Degree Duration

The minimum duration for completion of MS (CS) degree is two years (4-regular semester) and the maximum period is four years to complete MS (CS) 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 (Computer Science) 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 computer science fundamentals. The University expects its MS graduates to pursue careers either as ‘Computer Science Faculty Members’ or as ‘Software Development Managers’ in the industry.

Learning Outcomes:

1. Students will be able to possess advanced knowledge of Computer Science field
2. Students will be able to think creatively and critically; to solve non-trivial problems
3. Students will be able to use computing knowledge to develop efficient solutions for real life problems
4. Students will be able to design solutions and can conduct research related activities

1.7 Program Educational Objectives

1. Professional Competence

Graduates will demonstrate advanced knowledge and skills in computer science to effectively solve complex problems and contribute to academia, research, and the software industry.

2. Leadership and Innovation

Graduates will exhibit leadership qualities and drive innovation by developing intelligent, secure, and sustainable computing solutions using emerging technologies such as Artificial Intelligence.

3. Research and Lifelong Learning

Graduates will engage in research, critical inquiry, and lifelong learning to stay at the forefront of technological advancements and to contribute to the global knowledge economy.

4. Ethical and Social Responsibility

Graduates will uphold ethical practices and contribute to the betterment of society by applying computing solutions that address socio-economic and security challenges.

5. Empowerment and Inclusive Excellence

Graduates, will be empowered to participate in and lead interdisciplinary, collaborative initiatives, promoting inclusivity, equity, and diversity in the computing profession.

2. Area Covered in MS (CS) Program

Course Group	Credit Hours
Core Course	12
Mandatory Course	05
Elective	09
Research Thesis	6
Total Credit Hours	32

3. Scheme of Study of MS Computer Science in the light of HEC Curriculum

3.1 Core Courses (12 Credit Hours)

#	Code	Course Title	Credit hours
1	CS-701	Advanced Analysis of Algorithms	3
2	CS-702	Advanced Operating Systems	3
3	CS-703	Advanced Computer Architecture	3
4	CS-704	Theory of Computation	3
5	CS-705	Theory of Programming Languages	3

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

3.2 Mandatory Course (3 Credit Hours)

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

3.3 Elective Courses (9 Credit Hours)

#	Code	Course Title	Credit hours
1	CS-712	Advanced Requirement Engineering	3
2	CS-713	Advanced Global Software Development	3
3	CS-714	Software Quality Engineering	3
4	CS-715	Advanced Software Project Management	3
5	CS-721	Image Processing	3
6	CS-722	Computer Vision	3
7	CS-723	Deep Learning in Computer Vision	3
8	CS-731	Advanced Artificial Intelligence	3
9	CS-732	Deep Learning	3
10	CS-733	Natural Language Processing	3
11	CS-734	Machine Learning	3
12	CS-741	Advanced Computer Networks	3
13	CS-742	Security Management and Cryptography	3
14	CS-743	Distributed Systems	3
15	IT-701	Advance Ubiquitous Computing	3
16	IT-702	Advance Database Management System	3
17	IT-715	Advance Wireless Sensor Networks	3
18	IT-716	Advance Internet of Things	3
19	IT-717	Introduction to Software Defined Networking	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 (6 Credit Hours)

The department started MS (CS) degree containing research work of 6 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 (CS) 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 Holy Quran-I/ Ethics-I	Nil	1	Mandatory Course
		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 Holy Quran-II/ Ethics-II	Nil	1	Mandatory
	CS-711	Research Methodology	Nil	3	Mandatory Course
				13	
Third					
	CS-799	Research Thesis	Nil	6	MS Thesis
Fourth	CS-799	Research Thesis			
			Total	32	

5. Core Courses

#	Code	Course Title	Credit hours
1	CS-701	Advanced Analysis of Algorithms	3
2	CS-702	Advanced Operating Systems	3
3	CS-703	Advanced Computer Architecture	3
4	CS-704	Theory of Computation	3
5	CS-705	Theory of Programming Languages	3

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

6. Course Outline Core Courses

Advanced Analysis of Algorithms

MS-Computer Science

Course Code: **CS-701**

Credit Hours: 3

Pre-requisite: None

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:

- Prove the correctness of algorithms using rigorous mathematical techniques taught in this course.
- Analyze average and worst-case running times of given algorithm.
- 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.

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

Understand the definitions of P (polynomial time) and NP (nondeterministic polynomial time) problem classes. Learn how problems are reduced to one another in polynomial time to show relative complexity. Explore the concept

of NP-Complete problems, which are the hardest problems in NP. Study the foundational proof that SAT is NP-complete, introducing the concept of NP-completeness.

Recommended books/reference material

1. Introduction to Algorithms, 4th Edition 2022, by T. H. Cormen.
2. Souici, I., Mahamdioua, M., Jacques, S., & Ouahabi, A. (2025). Advanced genetic image encryption algorithms for intelligent transport systems. *Computers and Electrical Engineering*, 123, 110162.
3. Tomar, S., Tripathi, R., & Kumar, S. (2025). Comprehensive Survey of QML: From Data Analysis to Algorithmic Advancements. arXiv preprint arXiv:2501.09528.

Advanced Operating Systems

MS-Computer Science

Course Code: **CS-702**

Credit Hours: 3

Pre-requisite: None

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.
- Understand 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

1. Prakash, R. (2025). Advanced Operating Systems: List of Papers.
2. Comer, D. (2025). *Operating system design: The Xinu approach*. CRC Press.
3. Xing, T., Xiong, C., Wei, T., Sanchez, A., Ravindran, B., Balkind, J., & Barbalace, A. (2025, March). Stramash: A Fused-Kernel Operating System For Cache-Coherent, Heterogeneous-ISA Platforms. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2* (pp. 1172-1188).

Advanced Computer Architecture

MS-Computer Science

Course Code: **CS-703**

Credit Hours: 3

Pre-requisite: None

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.

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

1. Patronas, G., Terzenidis, N., Kashinkunti, P., Zahavi, E., Syrivelis, D., Capps, L., & Bakopoulos, P. (2025). Optical switching for data centers and advanced computing systems. *Journal of Optical Communications and Networking*, 17(1), A87-A95.
2. Veeramachaneni, V. (2025). Edge Computing: Architecture, Applications, and Future Challenges in a Decentralized Era. *Recent Trends in Computer Graphics and Multimedia Technology*, 7(1), 8-23.
3. Mandekar, U., Balpande, V. P., Borkar, P., & Aher, P. B. (2025). Introduction to High-Performance Computing Architectures. In *Parallel and High-Performance Computing in Artificial Intelligence* (pp. 1-21). Auerbach Publications.

Theory of Computation

MS-Computer Science

Course Code: **CS-704**

Credit Hours: 3

Pre-requisite: None

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, Introduction to 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. Finite Automata

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. Pushdown Automata

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 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

1. Oliveira, I. C. (2025). Meta-mathematics of computational complexity theory. *arXiv preprint arXiv:2504.04416*.
2. Jiang, Y., Zhang, X., Beer, M., Faes, M. G., Papadimitriou, C., & Zhou, H. (2025). First excursion probability of dynamical systems: A review on computational methods. *Mechanical Systems and Signal Processing*, 232, 112751.

Theory of Programming Languages

MS-Computer Science

Course Code: **CS-705**

Credit Hours: 3

Pre-requisite: None

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

- Understand the underlying theory of programming languages
- Analyze the problem and apply appropriate solution for the programming language.
- Analyze 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 The Lambda Calculus, Operational Semantics, Reduction Order,

Unit 5. Functions

Recursive Functions, Logic Programming, Inference Engine, Concurrency.

Recommended books/reference material

1. Poitras, E., Crane, B. G. C., Dempsey, D., Siegel, A. A., & Farion, C. (2025). Programming Language Learning in K-12 Education. Empowering STEM Educators With Digital Tools, 227-260
2. Hu, J. Z., & Pientka, B. (2025). A Dependent Type Theory for Meta-programming with Intensional Analysis. Proceedings of the ACM on Programming Languages, 9(POPL), 416-445.
3. Stassen, P., Møgelberg, R. E., Zwart, M. A., Aguirre, A., & Birkedal, L. (2025). Modelling Recursion and Probabilistic Choice in Guarded Type Theory. Proceedings of the ACM on Programming Languages, 9(POPL), 1417-1445.
4. Lee, M. P., Pryce, J. D., & Harrison, A. (2025). Prolog as a first programming language. WIT Transactions on Information and Communication Technologies.

7. Elective Courses

#	Code	Course Title	Credit hours
1	CS-711	Advanced Requirement Engineering	3
2	CS-712	Advanced Global Software Development	3
3	CS-713	Software Quality Engineering	3
4	CS-714	Advanced Software Project Management	3
5	CS-715	Image Processing	3
6	CS-716	Computer Vision	3
7	CS-717	Deep Learning in Computer Vision	3
8	CS-718	Advance Artificial Intelligence	3
9	CS-719	Deep Learning	3
10	CS-720	Natural Language Processing	3
11	CS-721	Machine Learning	3
12	CS-722	Advanced Computer Networks	3
13	CS-723	Security Management and Cryptography	3
14	CS-724	Distributed Systems	3

Note: Above mention 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 on the recommendation of departmental curriculum committee.

8. Course Outline Elective Courses

Advanced Requirement Engineering

MS-Computer Science

Course Code: **CS-712**

Credit Hours: 3

Pre-requisite: None

Objectives

The aim of course is enabling 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.

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.

Unit 2. Requirement elicitation techniques

Interviews, Workshops, Focus groups, Observations, Questionnaires, System interface analysis, User interface analysis, and 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 elaborate 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

1. Barbosa, L., Freire, S., Maciel, R. S., Mendonça, M., Kalinowski, M., Codabux, Z., & Spínola, R. (2025). Attributes of a great requirements engineer. *Journal of Systems and Software*, 219, 112200.
2. Lisda, L., Febrianto, D. C., & Kusumastuti, R. (2025). Designed a Waste Management Application by Applying Requirements Engineering Methods to Meet User Needs and Expectations. *Infotekmesin*, 16(1), 199-206.
3. Chen, X., Hu, X., Huang, Y., Jiang, H., Ji, W., Jiang, Y., ... & Zhang, L. (2025). Deep learning-based software engineering: progress, challenges, and opportunities. *Science China Information Sciences*, 68(1), 1-88.

Advanced Global Software Development

MS-Computer Science

Course Code: **CS-713**

Credit Hours: 3

Pre-requisite: None

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

- Understand the principles of software engineering in context of global software development.
- Evaluate and discuss the issues around global software development and techniques for managing distributed projects.
- Understand 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 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. Architectures and Coordination.

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. Existing Product Technologies

Reconfiguration of Existing Product Technologies, Identification of Coordination Requirements. Distributed Development Environments: Software configuration management, Awareness among Configuration Management.

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. 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. Zamir, S., Rehman, A., Mohsin, H., Zamir, E., Abbas, A., & Al-Yarimi, F. A. (2025). Integrating Pull Request Comment Analysis and Developer Profiles for Expertise-based Recommendations in Global Software Development. *IEEE Access*.
2. Mishra, D. (2025). Developing a knowledge-based perspective of coordination in global software development. *VINE Journal of Information and Knowledge Management Systems*, 55(2), 287-309.

Software Quality Engineering

MS-Computer Science

Course Code: **CS-714**

Credit Hours: 3

Pre-requisite: None

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.
Quality evaluation of software requirement document.

Unit 3. 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
Project management metrics, Software development process environmental metrics, product metrics, and activity metrics

Unit 4. 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

1. Farias, R. S., Ahmed, I., & de Almeida, E. S. (2025). What Makes a Great Software Quality Assurance Engineer?. *IEEE Transactions on Software Engineering*.
2. Burgess, C. J., & Jones, B. F. (2025). Should Software Quality Be A Major Issue When Teaching First Year Programming To Software Engineers?. *WIT Transactions on Information and Communication Technologies*, 7.
3. Georgiadou, E., Karakitsos, G., Sadler, C., & Stasinopoulos, D. (2025). An experimental examination of the role of re-engineering in the management of software quality. *WIT Transactions on Information and Communication Technologies*, 4.

Advanced Software Project Management

MS-Computer Science

Course Code: **CS-715**

Credit Hours: 3

Pre-requisite: None

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. 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.

Learning outcomes

- 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.

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

1. Alabi, A. A., Mustapha, S. D., & Akinade, A. O. (2025). Leveraging Advanced Technologies for Efficient Project Management in Telecommunications. *risk management (Cioffi et al., 2021; Lee et al., 2020)*, 17, 49.
2. Madauss, B. J. (2025). *Project Management: A comprehensive description of Theory and Practice*. Springer Nature.
3. Madanchian, M., & Taherdoost, H. (2025). Exploring the Landscape of Multicriteria Decision Making in Software Project Management: Trends, Challenges, and Future Directions. *EAI Endorsed Transactions on Scalable Information Systems*, 12(1).

Image Processing

MS-Computer Science

Course Code: **CS-721**

Credit Hours: 3

Pre-requisite: None

Objectives

To provide knowledge on basic concepts of image and its processing techniques
To provide knowledge on Enhancement, Restoration, Segmentation techniques
To provide hand on experience of image processing techniques using MATLAB.

Learning Outcomes:

- Students will gain knowledge on fundamental concepts of a digital image processing System.
- Students will develop skill of developing new algorithms in signal and image processing Applications.

- 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. Wang, J., Wang, S., & Zhang, Y. (2025). Deep learning on medical image analysis. *CAAI Transactions on Intelligence Technology*, 10(1), 1-35.
2. Kong, C., Luo, A., Wang, S., Li, H., Rocha, A., & Kot, A. C. (2025). Pixel-inconsistency modeling for image manipulation localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
3. Kim, J. W., Khan, A. U., & Banerjee, I. (2025). Systematic review of hybrid vision transformer architectures for radiological image analysis. *Journal of Imaging Informatics in Medicine*, 1-15.

Computer Vision

MS-Computer Science

Course Code: **CS-722**

Credit Hours: 3

Pre-requisite: None

Objectives

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. To develop a computer-based system with the capabilities of the human eye-brain system. To explore some of the basic principles and techniques which are currently being used in real-world computer vision systems.

Learning Outcomes

- Demonstrate a solid understanding of core concepts, theories, and methods in computer vision.
- Analyze and evaluate existing computer vision systems for their design, functionality, and effectiveness.
- Communicate effectively and collaborate in teams to design and implement a functional computer vision system.
- Critically review scientific literature to identify innovation and assess the practicality of computer vision methods, applying theoretical knowledge to real-world scenarios.

Course Outline

Unit 1.

Computer Vision Fundamentals

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.

Edge Detection and Shape Representation

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. 3D Vision and Camera Geometry

Camera models, 3D vision, Triangulation principle, Stereoscopy, Generating 3D images from 2D views.

Recommended books/reference material

1. Qu, H., Rahmani, H., Xu, L., Williams, B., & Liu, J. (2025). Recent advances of continual learning in computer vision: An overview. *IET Computer Vision*, 19(1), e70013.
2. Zhu, Y., & Shen, T. (2025, January). The role of machine learning in enhancing computer vision processing. In *The 2nd International scientific and practical conference “Innovative technologies for training and educating young people” (January 14–17, 2025) Boston, USA*. International Science Group. 2025. 309 p. (p. 115).
3. Upadhyay, A., Chandel, N. S., Singh, K. P., Chakraborty, S. K., Nandede, B. M., Kumar, M., ... & Elbeltagi, A. (2025). Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture. *Artificial Intelligence Review*, 58(3), 1-64.

Deep learning in computer vision

MS-Computer Science

Course Code: **CS-723**

Credit Hours: 3

Pre-requisite: None

Objectives

Design and implement an adequate data extraction and augmentation strategy
Implement Deep Learning algorithms with open source frameworks
Evaluate performances of the designed approaches in terms of common quality metrics.

Learning Outcomes:

- Build and train deep neural networks, implement vectorized neural networks and identify architecture parameters
- Use best practices to train and develop test sets, analyze bias/variance for building Deep Learning applications and apply optimization algorithms.
- Apply deep learning to image, video, and other 2D/3D data for visual detection and recognition tasks.

Course Outline

Unit 1. Color systems

Spatial sampling in images and video
Key methods of motion estimation in video

Unit 2. 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

Problem of training data, domain dependent data augmentation,
Optimization algorithms, transfer learning, deeper architectures ,
Information fusion with CNNs

Unit 5. Temporal aspects in Deep Learning

RNN, LSTM, 2D+t ConvNets , Applications case studies

Recommended books/reference material

1. Upadhyay, A., Chandel, N. S., Singh, K. P., Chakraborty, S. K., Nandede, B. M., Kumar, M., ... & Elbeltagi, A. (2025). Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture. *Artificial Intelligence Review*, 58(3), 1-64.
2. Xu, Y., Khan, T. M., Song, Y., & Meijering, E. (2025). Edge deep learning in computer vision and medical diagnostics: a comprehensive survey. *Artificial Intelligence Review*, 58(3), 1-78.

3. Ataei, S., Adibnazari, S., & Ataei, S. T. (2025). Data-driven Detection and Evaluation of Damages in Concrete Structures: Using Deep Learning and Computer Vision. *arXiv preprint arXiv:2501.11836*.

Advance Artificial Intelligence

MS-Computer Science

Course Code: **CS-731**

Credit Hours: 3

Pre-requisite: None

Objectives

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. Ability to apply knowledge representation, reasoning, and machine learning techniques to real-world problems.

Learning Outcomes

- 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.
- Ability to apply knowledge representation, reasoning, and machine learning techniques to real-world problems

Course Outline

Unit 1.

Introduction

Introduction to Artificial Intelligence, Foundation, The status of knowledge.

Unit 2.

Search Strategies

State Space Search, State Spaces, Environments and Problems, Production Systems, State Space Search Control, Game Search Control

Unit 3.

Knowledge Representation

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.**Semantics**

Knowledge Representation with Non-classical Logic, Uncertain epistemic status, imprecise assertional status, Taxonomies, ontologies, and knowledge bases

Unit 5.**Algorithms**

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
3. Force, A. A. T., Parasa, S., Berzin, T., Leggett, C., Gross, S., Repici, A., ... & Sharma, P. (2025). Consensus statements on the current landscape of artificial intelligence applications in endoscopy, addressing roadblocks, and advancing artificial intelligence in gastroenterology. *Gastrointestinal endoscopy*, 101(1), 2-9.
4. Ahmad, N., Yuan, J., & Zou, Y. (2025). One more step towards better stability of non-fullerene organic solar cells: Advances, challenges, future perspective, and era of artificial intelligence. *Energy & Environmental Science*.

Deep Learning

MS-Computer Science

Course Code: **CS-732**

Credit Hours: 3

Pre-requisite: None

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:

- Design convolution networks for handwriting and object classification from images or video.

- Design recurrent neural networks with attention mechanisms for natural language classification, generation, and translation.

Course Outline

Unit 1.

Introduction

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

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

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.

Transferring

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. Reinforcement Learning

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\
3. Wang, J., Wang, S., & Zhang, Y. (2025). Deep learning on medical image analysis. *CAAI Transactions on Intelligence Technology*, 10(1), 1-35.
4. Shah, M., & Sureja, N. (2025). A comprehensive review of bias in deep learning models: Methods, impacts, and future directions. *Archives of Computational Methods in Engineering*, 32(1), 255-267.
5. Wang, J., Wang, S., & Zhang, Y. (2025). Deep learning on medical image analysis. *CAAI Transactions on Intelligence Technology*, 10(1), 1-35.

Natural Language Processing

MS-Computer Science

Course Code: **CS-733**

Credit Hours: 3

Pre-requisite: None

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:

- 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.

- Develop a conversational agent that uses natural language understanding and generation.

Course Outline

Unit 1. NLP Fundamentals

Introduction to NLP, linguistics and NLP tasks, Python libraries for NLP, Text preprocessing and N-grams.

Unit 2. Sequence Modeling

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

Statistical machine translation, Machine translation: word alignment, parallel corpora, decoding, evaluation.

Unit 4. Advanced Parsing and Translation

Modern deep learning machine translation systems (phrase-based, syntactic), Syntax and parsing, co-reference resolution, Tree recursive neural networks for POS tagging.

Unit 5. Semantics and Dialogue Systems

Computational semantics, Question answering, Text summarization, Dialogue systems.

Recommended books/reference material

1. Joshi, A., Dabre, R., Kanojia, D., Li, Z., Zhan, H., Haffari, G., & Dippold, D. (2025). Natural language processing for dialects of a language: A survey. *ACM Computing Surveys*, 57(6), 1-37.
2. Feuerriegel, S., Maarouf, A., Bär, D., Geissler, D., Schweisthal, J., Pröllochs, N., ... & Van Bavel, J. J. (2025). Using natural language processing to analyse text data in behavioural science. *Nature Reviews Psychology*, 1-16.

3. Lane, H., & Dyshel, M. (2025). *Natural language processing in action*. Simon and Schuster.

Machine Learning

MS-Computer Science

Course Code: **CS-734**

Credit Hours: 3

Pre-requisite: None

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:

- Understand a wide variety of learning algorithms.
- Understand how to evaluate models generated from data.
- 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

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. Components

Part II (Four Components of Machine Learning Algorithm (Hypothesis, Loss Functions, Derivatives and Optimizatio Algorithms), Gradient Descent, Stochastic Gradient Descent

Unit 3. Regression

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. Models

Gaussian mixture models, EM algorithm, etc.; Reinforcement learning; tuning model complexity; Bias-Variance Tradeoff;

Unit 5. Search Strategies

Grid Search, Random Search; Evaluation Metrics; reporting predictive performance

Recommended books/reference material

1. Pattern Recognition and Machine Learning (Information Science and Statistics) by Christopher M. Bishop, 2016 Pattern Recognition & Machine Learning, 1st Edition.
2. Bracco, A., Brajard, J., Dijkstra, H. A., Hassanzadeh, P., Lessig, C., & Monteleoni, C. (2025). Machine learning for the physics of climate. *Nature Reviews Physics*, 7(1), 6-20.
3. Applied Machine Learning and AI for Engineers: Solve Business Problems That Can't Be Solved Algorithmically by Jeff Prosise, 2022
4. Meng, C., Griesemer, S., Cao, D., Seo, S., & Liu, Y. (2025). When physics meets machine learning: A survey of physics-informed machine learning. *Machine Learning for Computational Science and Engineering*, 1(1), 1-23.

Advanced Computer Networks

MS-Computer Science

Course Code: **CS-741**

Credit Hours: 3

Pre-requisite: None

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**

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

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

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. Pramudito, D. K., & Fatchan, M. (2025). Optimizing Image Recognition in Low-Light Environments through Advanced Computer Vision Algorithms. *The Journal of Academic Science*, 2(1), 278-289.
2. Dandamudi, S. R. P., Sajja, J., & Khanna, A. (2025). AI transforming data networking and cybersecurity through advanced innovations. *International Journal of Innovative Research in Computer Science and Technology*, 13, 42-49.
3. Mukesh, V. (2025). A Comprehensive Review of Advanced Machine Learning Techniques for Enhancing Cybersecurity in Blockchain Networks. *Journal ID*, 8736, 2145.

Security Management and Cryptography

MS-Computer Science

Course Code: **CS-742**

Credit Hours: 3

Pre-requisite: None

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, and 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.
2. Pramudito, D. K., & Fatchan, M. (2025). Optimizing Image Recognition in Low-Light Environments through Advanced Computer Vision Algorithms. *The Journal of Academic Science*, 2(1), 278-289.
3. Dandamudi, S. R. P., Sajja, J., & Khanna, A. (2025). AI transforming data networking and cybersecurity through advanced innovations. *International Journal of Innovative Research in Computer Science and Technology*, 13, 42-49.

Distributed Systems

MS-Computer Science

Course Code: **CS-743**

Credit Hours: 3

Pre-requisite: None

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:

- 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.
- 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

Unit 2. TCP/IP, UDP, Remote Method Invocations

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
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
Soap 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

1. Shah, M., Hazarika, A. V., Malhotra, M., Patil, S. C., & Mohanty, J. (2025). Bridging Emotions and Architecture: Sentiment Analysis in Modern Distributed Systems. *arXiv preprint arXiv:2503.18260*.
2. Lu, R., Lu, Y., Jiang, Y., Xue, G., & Huang, P. (2025). {One-Size-Fits-None}: Understanding and Enhancing {Slow-Fault} Tolerance in Modern Distributed Systems. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)* (pp. 359-378).
3. Shah, M., & Hazarika, A. V. (2025). An In-Depth Analysis of Modern Caching Strategies in Distributed Systems: Implementation Patterns and Performance Implications. *International Journal of Science and Engineering Applications (IJSEA)*, 14(1), 9-13.

Advance Ubiquitous Computing

MS-Computer Science

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

Mark Weiser’s vision of ubiquitous computing is seamless, invisible tech in daily life. It’s context-aware, adaptive, and used in healthcare, transport, homes, retail, and education—different from mobile (portable) and pervasive (widespread) computing.

Unit 2: Context-Awareness and Smart Environments

Context is information about a situation; context modeling structures it. Systems acquire, reason, and manage this data to enable context-aware, adaptive behavior in smart spaces. User modeling allows personalized experiences.

Unit 3: System Architecture and Middleware

Ubiquitous systems use layered architectures with middleware like Context Toolkit or OpenIoT for context-aware computing. They support service discovery, messaging, and device interoperability, with edge, fog, and cloud integration for efficient data processing.

Unit 4: Sensing, Communication, and Device Integration

Ubiquitous systems use sensors—wearable, ambient, and mobile—for data collection. Embedded platforms like Arduino, Raspberry Pi,

and ESP32 support these systems. They communicate via protocols like Bluetooth, ZigBee, MQTT, CoAP, and LoRa, with a focus on energy efficiency and mobility management.

Unit 5: Security, Privacy, and Ethical Issues

Ubiquitous computing faces threats like data breaches and surveillance. Key protections include anonymization, user consent, and ethical data collection. Case studies in tracking and health monitoring highlight privacy and security challenges.

Unit 6: Emerging Trends and Case Studies

Ubiquitous AI uses machine learning for context prediction, while BCIs enable direct brain-system interaction. Digital twins support real-time simulation. Case studies in smart cities, agriculture, and assisted living show impact. Research trends focus on personalization, ethics, and intelligent automation.

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.

Advance Cloud Computing

MS - Computer Science

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

Cloud computing evolved from mainframes to on-demand internet services. It offers scalability, flexibility, and cost-efficiency through IaaS, PaaS, and SaaS models. Deployment types include public, private, hybrid, and community clouds. Major providers include AWS, Azure, and GCP.

Unit 2: Cloud Architecture and Virtualization

Cloud architecture includes hardware, virtualization, and management layers. Virtualization uses hypervisors (Xen, KVM, VMware) and containers (Docker, LXC). Resources are provisioned and scaled elastically. Orchestration tools like OpenStack, Kubernetes, and Terraform manage infrastructure efficiently.

Unit 3: Cloud Storage and Data Management

Cloud storage includes block, object, and file models. Distributed file systems like HDFS and GFS support large data. Data consistency balances with availability (CAP theorem), often using eventual consistency. NoSQL databases like MongoDB and Cassandra handle cloud data flexibly.

Unit 4: Cloud Application Development and Platforms

PaaS provides platforms for developing cloud-native apps. Serverless computing (e.g., AWS Lambda) runs functions on demand. Microservices and 12-factor apps enable scalable design. DevOps uses CI/CD pipelines for fast, automated cloud deployment.

Unit 5: Security and Privacy in Cloud Computing

Cloud security involves managing threats and risks through identity and access management (IAM), encrypting data at rest and in transit, and ensuring compliance with regulations like GDPR and HIPAA. Multi-tenancy requires strong isolation to protect users.

Unit 6: Emerging Trends and Case Studies

Edge/fog computing processes data locally. Multi/hybrid clouds offer flexible management. Green cloud promotes sustainability. AI/ML services enhance apps. Used in healthcare, finance, education, government. Research targets cloud federation and standards.

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.

Advance Wireless Sensor Networks

MS - Computer Science

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

Wireless Sensor Networks (WSNs) use distributed sensor nodes with limited energy and memory to monitor environments. They differ from traditional networks by focusing on energy efficiency and low latency. Nodes include sensors, processors, and communication units, designed for scalable, reliable data collection.

Unit 2: Medium Access Control (MAC) Protocols

MAC in WSNs faces challenges like energy waste and collisions. Energy-efficient protocols like S-MAC, T-MAC, and B-MAC use adaptive duty cycling to save power. Access methods include TDMA (scheduled) and contention-based schemes for communication.

Unit 3: Routing Protocols

WSN routing tackles energy and reliability issues. Flat routing treats all nodes equally; hierarchical uses clusters; location-based relies on node positions. Data-centric routing focuses on data, not addresses.

Unit 4: Localization, Time Synchronization & Coverage

WSNs use GPS-free localization, TPSN/RBS for synchronization, and models for coverage and connectivity. Mobility and deployment improve network efficiency.

Unit 5: Power Management and Energy Harvesting

Energy-efficient communication in WSNs uses sleep/wake scheduling, energy-aware routing and MAC protocols, plus energy harvesting. Battery models help predict node lifetime.

Unit 6: Security and Fault Tolerance in WSNs

WSNs face threats like attacks on data and routing. Lightweight cryptography and key management secure resource-limited nodes. Secure routing, data aggregation, and fault detection ensure network reliability and recovery.

Unit 7: WSN Platforms, Programming & Simulation

WSNs use sensor OS like TinyOS and Contiki, with specific programming models. Simulators like NS-3, Cooja, and TOSSIM help test networks. Performance is evaluated by metrics like delay and energy. Applications include environmental monitoring, smart agriculture, and military systems.

Unit 8: Advanced Topics and Research Trends

WBANs monitor body data and integrate with IoT for smart health. Cognitive WSNs adapt to environments. Mobile sinks and delay-tolerant designs improve data delivery. WSNs support smart cities and edge computing applications.

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.

Advance Internet of Things

MS - Computer Science

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

IoT links devices for smart automation, evolving from embedded systems. It's scalable and connected, with layers for sensing, networking, and applications. Unlike traditional or cyber-physical systems, IoT focuses on broad connectivity and data sharing.

Unit 2: IoT Hardware and Device Platforms

Microcontrollers like Arduino, ESP32, and Raspberry Pi control sensors and actuators. They use GPIO, I2C, SPI, and UART for communication and manage power efficiently.

Unit 3: Communication and Networking in IoT

IoT uses IPv6 and 6LoWPAN for networking, with ZigBee, BLE, LoRaWAN, and NB-IoT at the data link layer. MQTT, CoAP, and AMQP handle transport. Edge and fog computing bring processing closer to devices.

Unit 4: Data Acquisition and Processing

Sensors convert analog signals to digital, then process data before real-time transmission. Processing can happen locally (edge) or remotely (cloud) for efficiency.

Unit 5: IoT Platforms and Cloud Integration

Cloud IoT services like AWS IoT, Google IoT Core, and Azure IoT Hub manage device data. They handle data ingestion, storage, and visualization via dashboards, using RESTful APIs and webhooks for integration.

Unit 6: Security and Privacy in IoT

IoT devices face vulnerabilities like weak security and firmware attacks. Encryption, secure communication, and identity management help protect data. Blockchain adds trust. Cases include DDoS attacks and compromised firmware.

Unit 7: Industrial and Research Applications of IoT

IoT powers smart homes, cities, and transport systems, enabling automation and efficiency. It drives Industry 4.0 through IIoT, supports healthcare with wearables, and boosts yield in precision agriculture.

Unit 8: Recent Trends and Challenges

IoT combines AI with IoT for smarter decisions. Key challenges include interoperability, standards, energy efficiency, and ethical, legal, and environmental concerns. Future research focuses on secure, scalable, and sustainable IoT systems.

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.

Introduction to Software Defined Networks

MS - Computer Science

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

SDN separates the control plane from the data plane, unlike traditional networking. It offers centralized management, flexibility, and programmability, but faces challenges like scalability and security. SDN architecture includes application, control, and infrastructure layers, with both centralized and distributed control options.

Unit 2: Open Flow and Southbound Interfaces

Open Flow enables SDN by linking switches and controllers. It uses flow tables with match-action rules for packet handling. Versions evolve for better control and features.

Unit 3: SDN Controllers

SDN controllers (e.g., Ryu, ONOS) manage networks via southbound and northbound APIs. Centralized or clustered, they ensure control, scalability, and reliability.

Unit 4: Northbound APIs and Network Programming

Northbound APIs let apps control the network; often RESTful with SDKs. Programmability via Python/Java enables policy enforcement and intent-based networking for automation.

Unit 5: SDN Applications and Use Cases

SDN enables traffic engineering, load balancing, and NFV for flexible networking. It enhances security with firewalls and DDoS detection, and is widely used in data centers, ISPs, clouds, and enterprises.

Unit 6: SDN Virtualization and Emulation

Network slicing and virtualization create isolated networks. Overlay networks use tunneling (VXLAN, GRE). SDN is tested with tools like Mininet, GNS3, NS-3 and supports container platforms like Docker and Kubernetes.

Unit 7: Security and Fault Tolerance in SDN

SDN faces threats across controllers and switches. Secure communication, strong authentication, and authorization protect the network. Controller failure requires recovery plans, supported by trust and verification frameworks.

Unit 8: Research Trends and Future of SDN

SDN powers 5G and SD-WAN for flexible networks. AI/ML enhances SDN automation. Open standards and open-source tools drive innovation. Future trends focus on scalability, security, and integration.

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.

9. Mandatory Courses

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

10.Course Outline Mandatory Courses

Research Methodology

MS-Computer Science

Course Code: **CS-711**

Credit Hours: 3

Pre-requisite: None

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

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. Testing

- 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. Methods

- Data collection methods - primary and secondary sources, Types of data analysis methods
- Survey Research, Sampling Methods, Survey Study Designs, Case Studies.

Unit 5. Introduction to Mixed Methods Research, Study Designs and Methods

- Writing research papers, purpose, nature and evaluation, content and format, Research Presentations
- The Art of Scientific and Technical Writing

Recommended books/reference material

1. Sevón, E., Mustola, M., Siippainen, A., & Vlasov, J. (2025). Participatory research methods with young children: A systematic literature review. *Educational review*, 77(3), 1000-1018.
2. Moyser, G., & Wagstaffe, M. (Eds.). (2025). *Research methods for elite studies*. Taylor & Francis.
3. Huang, L., Zan, J., Lv, K., & Zhao, X. (2025). A systematic review of mixed methods research in tourism and hospitality. *Journal of Hospitality and Tourism Management*, 63, 163-176.

Understanding of Quran/ Fehm e Quran

Course Title: Understanding of Quran - I

Course Code: UHQ-101

Credit Hours: 1

Contact Hours: 3 per week

Weeks: 15-16 (45-48 hours)

Pre-Requisites: Nill

Course Objectives:

This course will enable the students to understand the Holy Quran by learning Arabic in a simple and easy way. Through relating their basic Urdu vocabulary with Quran, and without engaging them in complex Arabic Grammar, Sarf or Balagha. Students will gradually build the skills to read and understand Quranic verses directly—without needing translation. The course focuses on making Arabic accessible for everyone, even those with no prior background.

Learning Outcomes:

This course will enable the students to:

- Develop the ability to understand basic words of the Quran, phrases and sentences that do not contain verbs (unit 1 to 5 of Muallim ul Quran Book) and then sentences having present tense (first half of unit 6 of Muallim ul Quran Book).
- Acquire a strong foundation for understanding long verses of the Quran with clarity.
- Comprehend Quranic vocabulary, particles (operative & non operative particles), compounds (Adjective & Possessive compound), pronouns (singular & plural) and types of plurals through hundreds of Quranic sentences.
- Recognize and understand different styles of Quranic sentences, including nominal sentence, emphatic sentence, double emphatic sentence, negative sentence, interrogative sentence, oath –based sentences.
- Strengthen understanding of fundamental Quranic linguistic styles, expressions and idioms.
- Understand at least 30 to 40 % of each page of the Holy Quran.

Course Outline:

Unit 1. Simple Quranic Words

Muallim ul Quran: Book 1: Unit: Lesson 1-19

Unit 2. Words that comes before (The simple Quranic words)

Muallim ul Quran: Book 1: Unit#2: Lesson 1-13

Unit 3. Quranic words in terms of their relationship and characteristics

Muallim ul Quran: Book 1: Unit#3: Lesson 1-10

Unit 4. Singular Pronouns

Muallim ul Quran: Book 2: Unit#4: Lesson 1-17

Unit 5. Singular pronouns & Demonstrative Pronouns

Muallim ul Quran: Book 2: Unit#5: Lesson 1-23

Unit 6. Present Tense

Muallim ul Quran: Book 3: Unit#6: Lesson 1-5

Reference Books:

1- بشير، دكتور. عبید الرحمن - معلم القرآن - اسلام آباد: دار العلم، ۲۰۲۴. جلد ۱، ۲ & ۳

Ethics (in lieu of Understanding of Quran)

Course Title: Ethics-I

Course Code: ETH-103

Credit hours: 1

Contact Hours: 3 per week

Weeks: 15-16 (45-48 hours)

Pre-Requisites: Nil

Objectives

Upon successful completion of the course, students will be able to know the basics of ethics, & know how to develop an ethical approach. Also, analyze the importance of ethics and ethical values for a human being. And identify the Semitic & Non-Semitic Religions. Moreover, develop the concept of Ethics in Semitic & Non-Semitic religions. Also, analyze the similarities and differences between ethical teachings of various religions.

Learning Outcomes:

- Students will be able to demonstrate their understanding about the concept of ethics & ethical approach.
- Analyze the difference between the concept of Ethics in Semitic and Non-Semitic religions.

Course Outline:

Unit 1. Introduction and Importance of Ethics & Basic Ethical Principles

Definition of Ethics, Importance of Ethics, Beneficence, Non-maleficence, Autonomy, Justice as basic ethical behaviors.

Unit 2. Theories of Moral Standards

Moral Standards as Law, Moral Standards as Pleasure, Moral standards as Perfection. (Theories of various philosophers such as Kant, Spencer, John S. Mackenzie etc)

Unit 3. The concept of Ethics in Major Religions- Christianity, Judaism

Introduction of the Religion, Its Basic Concepts and major ethical teachings.

Unit 4. The concept of Ethics in Major Religions- Hinduism & Buddhism

Introduction of the Religion, Its Basic Concepts and major ethical teachings.

Unit 5. The concept of Ethics in Major Religions- Islam

Introduction of the Religion, Its Basic Concepts and major ethical teachings.

References/Recommended Books:

Books

1. Davis, N.K. (2023). *Elements of Ethics*, Legare Street Press.
2. Edmonds,D. (2019). *Ethics and the Contemporary World*, Taylor & Francis,
3. Hastings, J. Alexander, J. Gray S.L. (2023). *The Encyclopedia of Religion and Ethics*, Creative Media Partners, LLC
4. Robinson, D.Garratt C. (2008). *Introducing Ehics-a Graphic Guide*, Icon Books ltd.Uk.
5. اخلاق اور فلسفہ اخلاق، سیوہاروی، حفظ الرحمن، ندوة المصنفین، دہلی، ۱۹۷۶
6. دنیا کے بڑے مذاہب، فاروقی، عماد الحسن، مکتبہ جامعہ لمیٹڈ، دہلی، ۱۹۸۶

Websites & Articles

1. Articles on Ethics www.infotraccollege.com
2. Brown. (2017). The ethical dimensions of Global Environmental Issues. American Academy of Arts and Sciences. Retrieved July 7 ,2017 from amacad.org
3. Business Ethics Magazine www.business-ethics.com/annual.htm
4. Bussmann, J. (2016). Top 9 ethical Issues in artificial intelligence. Retrieved 7 July2017 from world Economic Forum weforum.org
5. Ethics and Religion, Shefali Narange <http://www.paperpublications.org/>
6. <http://www.jstor.com/stable/resrep12025.7>
7. <http://www.learnwell.org/ethicalproblems.htm>
8. Institute for Global Ethics www.globalethics.org
9. Klimes, Rudolf e. (2000). Ethical Problems.
10. Ten Ethical Problems of the Present World <https://www.lifepersona.com/10-ethical-problems-of-the-present-world>
11. Ward, Stephen J.A (March 2005). “Philosophical Foundations for Global Journalism Ethics” *Journal of Mass Media Ethics*:20(1)

Understanding of Quran/ Fehm e Quran

Course Title: Understanding of Quran - II

Course Code: UHQ-102

Credit Hours: 1

Contact Hours: 3 per week

Weeks: 15-16 (45-48 hours)

Pre-Requisites: Nil

Course Objectives:

This course will enable the students to understand the Holy Quran by learning Arabic in a simple and easy way. Through relating their basic Urdu vocabulary with Quran, and without engaging them in complex Arabic Grammar, Sarf or Balagha. Students will gradually build the skills to read and understand Quranic verses directly—without needing translation. The course focuses on making Arabic accessible for everyone, even those with no prior background.

Learning Outcomes:

This course will enable the students to:

- Directly comprehend hundreds of Quranic sentences & verses.
- Understand at least 80 to 85 % of each page of the holy Quran.
- Understand common verses across different Quranic topics.
- Achieve proficiency in the basic and advance linguistic aspects of the Arabic language.
- Understand the difference between Quranic verbs in various forms, such as present, past and imperative.
- Develop the ability to understand long verses of the holy Quran independently and then comprehend their interpretation.

Course Outline:

Unit 1.	<u>Past Tense</u> Muallim ul Quran: Book 4: Unit#7: Chapter 1: Lesson 1-9
Unit 2.	<u>Past Tense</u> Muallim ul Quran: Book 4: Unit#7: Chapter 2: Lesson 1-7
Unit 3.	<u>Past Tense</u> Muallim ul Quran: Book 4: Unit#7: Chapter 3: Lesson 1-9
Unit 4.	<u>Past Tense</u> Muallim ul Quran: Book 5: Unit#7: Chapter 4: Lesson 1-5
Unit 5.	<u>Past Tense & Passive Voice</u> Muallim ul Quran: Book 5: Unit#7: Chapter 5 - 9
Unit 6.	<u>imperative Verbs for Singulars</u> Muallim ul Quran: Book 6: Unit#8: Lesson 1-8

Reference Books:

- 1- بشير، دكتور. عبید الرحمن - معلم القرآن - اسلام آباد: دار العلم، ۲۰۲۴. جلد ۳، ۴، ۵، ۶

Ethics (in lieu of Understanding of Quran)

Course Title: Ethics-II

Course Code: ETH-104

Credit hours: 1

Contact Hours: 3 per week

Weeks: 15-16 (45-48 hours)

Pre-Requisites: Nil

Objectives

Upon successful completion of the course, students will be able to have better understanding of ethics in various walks of life like in workplace, in business, journalism etc. Moreover, they will be able to be familiarize with Contemporary World's Ethical Issues. And, try to develop answers about them.

Learning Outcomes:

- Students will be able to demonstrate their understanding about the importance of ethics in their lives today.
- Identify Contemporary World's Ethical Issues.
- Create problem solving for these contemporary Ethical issues.

Course Outline:

Unit 1. Professional Ethics & Bioethics

Meaning of Work/Professional Ethics, Professional Ethics and Human Values, Elements of professional Ethics:

Person attitude, Beliefs about work, Positive Attributes (honesty, integrity, dedication, commitment, responsibility, interpersonal relationship)

Genetic engineering, Organ Donation, Cloning, Head Transplant

Unit 2. Ethical Conduct in Business Ethics & it's important

Types of Business Ethics, Common Business Ethical Challenges

Conflict of interest, honesty & integrity, whistle blowing, loyalty vs truth

Unit 3. Journalism

Principles of Ethical Journalism:

Truth & Accuracy, Independence, Fairness & Impartiality, Humanity, Public Accountability

Unit 4 Environmental Ethics

Core Principles of environmental ethics:

Justice & sustainability, Sufficiency & compassion, Solidarity & public participation,

Pollution prevention.

Current issues of environmental ethics:

Pollution, The depletion of natural resources, Loss of biodiversity, Destruction of ecosystem, Global climate change

Unit 5. Social Inequalities

Meaning of Social Inequalities, Issues related to social inequality

Difference in source of income, Health care, Freedom of speech, Gender inequality, Toxic workplace culture, Education, Political representation & participation

Unit 6. Social Media Ethics

Facebook, YouTube, WhatsApp, Instagram, TikTok, Snapchat, LinkedIn, X
Current Ethical issues in social media:

Plagiarism, Fake news, Harassment, Photo ethics, Post without caution, inappropriate behavior privacy

References/Recommended Books:

Books

1. Davis, N.K. (2023). *Elements of Ethics*, Legare Street Press.
2. Edmonds, D. (2019). *Ethics and the Contemporary World*, Taylor & Francis,
3. Hastings, J. Alexander, J. Gray S.L. (2023). *The Encyclopedia of Religion and Ethics*, Creative Media Partners, LLC
4. Robinson, D. Garratt C. (2008). *Introducing Ethics-a Graphic Guide*, Icon Books Ltd.Uk.
5. اخلاق اور فلسفہ اخلاق، سیوہاروی، حفظ الرحمن، ندوة المصنفین، دہلی، ۱۹۷۶
6. دنیا کے بڑے مذاہب، فاروقی، عماد الحسن، مکتبہ جامعہ لمیٹڈ، دہلی، ۱۹۸۶

Websites & Articles

1. Articles on Ethics www.infotracs-college.com
2. Brown. (2017). The ethical dimensions of Global Environmental Issues. American Academy of Arts and Sciences. Retrieved July 7 ,2017 from amacad.org
3. Business Ethics Magazine www.business-ethics.com/annual.htm
4. Bussmann, J. (2016). Top 9 ethical Issues in artificial intelligence. Retrieved 7 July 2017 from world Economic Forum weforum.org
5. Ethics and Religion, Shefali Narange <http://www.paperpublications.org/>
6. <http://www.jstor.com/stable/resrep12025.7>
7. <http://www.learnwell.org/ethicalproblems.htm>
8. Institute for Global Ethics www.globalethics.org
9. Klimes, Rudolf e. (2000). Ethical Problems.
10. Ten Ethical Problems of the Present World <https://www.lifepersona.com/10-ethical-problems-of-the-present-world>
11. Ward, Stephen J.A (March 2005). “Philosophical Foundations for Global Journalism Ethics” *Journal of Mass Media Ethics*:20(1)