COURSE DESCRIPTIONS

Undergraduate Programs

Irfan Ud-Din

List of Courses

CS111 - Programming Fundamentals 2

CS112- Object Oriented Programming 3

CS113- Digital Logic Design 4

CC211- Data Structures 5

CS212 - Database Systems 6

CS213 - Software Engineering 7

CS214 – Design and Analysis of Algorithms 8

CS215 - Operating Systems 9

CS311 - Computer Networks 10

CS312 - Artificial Intelligence 11

CS313 - Theory of Automata 12

CS411 - Information Security 13

CS412 - Computer Organization and Architecture 14

Specialization – General Computer Science (Honors) 15

CS321 - Web Technologies 16

CS322 - Advanced Programming 17

CS323 - Web Engineering 18

CS421 - Machine Learning 19

CS422 - Mobile Application Development 20

CS423 - Internet of Things and Its Applications 21

CS424 - Network Security 22

CS425 – Digital Image Processing [TODO] 23

Specialization – Software Engineering 24

CS331- Software Requirement Engineering 25

CS332 - Software Design and Architecture 26

CS333 - Human Computer Interaction 27

CS431 – Formal Method and Design [todo] 28

CS433 - Software Project Management 29

CS434 - Parallel and Distributed Computing 30

CS435 - Software Quality and Testing 31

Specialization – Artificial Intelligence 32

CS431 - Programming for AI 33

CS343 - Knowledge Representation & Reasoning 34

CS441 - Computer Vision 35

CS442 - Natural Language Processing 36

CS443 - Deep Learning 37

CS444 – Evolutionary Computing & Swarm Intelligence 38

CS445 - Reinforcement Learning 39

Specialization – Cyber Security 40

CS351 - Wireless and Mobile Security 41

CS352 - Secure Software Design and Development 42

CS353 - Digital Forensics and Incident Response 43

CS452 – Applied Cryptography [todo] 43

CS453 – Usable Security and Policy [todo] 43

CS454 – Cloud Security [todo] 44

CS455 – AI for Cyber Security [todo] 44

Fundamental Computing or Mathematical Courses 45

CS101 – Application of ICT [todo] 46

CS102 – Applied Mathematics for Computing I 47

CS103 – Applied Mathematics for Computing II 48

Specialization – Professional Courses 49

CS496 – Professional Certification 50

CS497 – Internship 51

CS111 - Programming Fundamentals

Credit Hours: 4(3+1) **Prerequisites:** None
Course Category: Computer Science Core Course

Description
This course furnishes students with a comprehensive foundation in fundamental concepts essential to any standard programming language. It covers key programming principles in structured ensuring students grasp these concepts. Moreover, students will acquire hands-on experience utilizing the C++ programming language, along with insights into basic understanding about the object-oriented program development.

Textbook(s)
- Starting out with Programming Logic & Degins, 4th Edition, Tony Gaddis, Pearson (2018), ISBN:978-0134801155

- Object Oriented Programming in C++ by Robert Lafore, Sams, (2001), ISBN: 978-0672323089

Other supplemental materials
- C++ online tutorial - https://www.w3schools.com/cpp/
- C++ Online Compiler - https://www.programiz.com/cpp-programming/online-compiler/

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Demonstrate proficiency in writing, debugging, and executing basic programs using programming languages such as C or Python.	C2 1,2,3
CLO 2	Explain core programming concepts including variables, control structures, functions, and data types.	C3 1,2
CLO 3	Develop simple algorithms to solve computational problems.	C3 3,4,5, 6, 8
CLO 4	Understand the basics of Object-Oriented Programming.	C4 1,2,3, 5

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to problem solving, a brief review of Von-Neumann architecture
2	Introduction to programming, role of compiler and linker,
3	introduction to algorithms and transformation into programs
4	Basic data types and variables, input/output constructs,
5	Arithmetic, comparison and logical operators
6	Conditional statements and execution flow for conditional statements
7	Repetitive statements and execution flow for repetitive statements
8	Arrays and their memory organization, multi-dimensional Arrays
9	Introduction to modular programming, function definition and calling
10	Stack rolling and unrolling, vectors
11	string and string operations
12	Pointers/references,
13	Static and dynamic memory allocation
14	File I/O operations
15	Structures. Basics about Object Oriented Programming
16	Sample Project (Case Study)

CS112- Object Oriented Programming

Credit Hours: 4(3+1) **Prerequisites:** CS111
Course Category: Computer Science Core Course

Description
The course aims to focus on object-oriented concepts, analysis and software development. The basic concept of OOP is covered in this course.

Textbook(s)
- Paul Deitel, Harvey Deitel. Java How to Program, Early Objects, 11th Edition, Peason, (2017), ISBN: 978-0134743356
- Nick Samoylov. Learn Java 17 Programming: Learn the fundamentals of Java Programming with this updated guide with the latest features, 2nd Edition, Packet Publishing, (2022),ISBN: 978-1803241432

Other supplemental materials
- Online Tutorial – https://www.w3schools.com/cpp/cpp_oop.asp

- Online Compiler - https://ideone.com/xA0en6

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand principles of object-oriented paradigm	C2 1, 10
CLO 2	Identify the objects & their relationships to build object-oriented solution	C3 2,6
CLO 3	Model a solution for a given problem using object-oriented principles	C3 3,4, 8, 10
CLO 4	Examine an object-oriented solution	C4 5,6, 8

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to object-oriented design, history and advantages of object-oriented design
2	Introduction to object-oriented programming concepts, Basic UML, Primitive vs Reference Types, Memory Models, Initializing Objects, Encapsulation, Constructors and Method overloading, Default and no argument constructors.
3	Variable declaration, data types, input output operations
4	Objects and classes, Access modifiers, Data encapsulation, constructors, destructors
5	Static data members & functions, const vs non-const functions
6	Identification of classes and their relationships, memory models, Composition, aggregation
7	Inheritance, class hierarchies, super classes and subclasses
8	inheritance, multiple inheritance, constructors in subclasses, Function overloading, Operator overloading
9	Exception Handling
10	Polymorphism – abstract and concrete classes
11	Polymorphism - constructor overloading, interfaces. Case Study
12	Strings and string operations
13	Graphical User Interfaces, 2D graphics
14	Databases with GUI
15	File Handling and working with data
16	Collections

CS113- Digital Logic Design

Credit Hours: 4(3+1) **Prerequisites:** None
Course Category: Computer Science Core Course

Description
The course introduces the concept pf digital logic, gates and the digital circuits. Further, it focuses on the design and analysis combinational and sequential circuits. It also serves to familiarize the students with the logic design of basic computer hardware components.

Textbook(s)
- Thomas Floyd. Digital Fundamentals 11th Edition, Pearson, (2014), ISBN: 978-0132737968
- Stephen Brown and Zvonko Vranesic. Fundamentals of Digital Logic with Verilog Design, Mc Graw Hill, (2017), ISBN: 978-0070667242

Other supplemental materials
- Draw Gates online - https://app.creately.com/d/JGzTzOr0LGa/edit

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Acquire knowledge related to the concepts, tools and techniques for the design of digital electronic circuits	C2 1,2, 10
CLO 2	Demonstrate the skills to design and analyze both combinational and sequential circuits using a variety of techniques	C3 3,4, 5, 10
CLO 3	Apply the acquired knowledge to simulate and implement small-scale digital circuits	C3 4,5
CLO 4	Understand the relationship between abstract logic characterizations and practical electrical implementations.	C2 3

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to Course, Number Systems
2	Binary Arithmetic, binary codes, error detection and correction (Parity code, Hamming code)
3	Boolean Algebra: Laws, Theorems, and Simplification
4	Logic gates & circuit construction (AND, OR, NOT, NAND, NOR, XOR)
5	Simplification Methods (K-Map, Quinn Mc-Cluskey method)
6	Combinational Circuits and designs I: Adders, Subtractors, Comparators
7	Combinational Circuits and designs II: Decoders, Encoders, Mux, DeMux
8	Sequential Circuits and designs: Flip Flops and Latches
9	Asynchronous and Synchronous circuits
10	Shift Registers
11	Counters
12	Triggered devices & its types
13	Arithmetic Circuits Arithmetic Circuits: Ripple-Carry, Carry Look-Ahead Adders, ALU Basics
14	Memory Elements
15	Introduction Programmable Logic Devices (CPLD, FPGA)
16	Introduction to HDL/VHDL

CC211- Data Structures

Credit Hours: 4(3+1) **Prerequisites:** CS111
Course Category: Computer Science Core Course

Description
The course is designed to teach students structures and schemes, which allow them to write programmer to efficiently manipulate, store, and retrieve data. Students are exposed to the concepts of time and space complexity of computer programs.

Textbook(s)
- Mark Weiss. Data Structures and Algorithm Analysis in Java 3rd Edition, Pearson, (2011), ISBN: 978-0132576277
- Adam Drozdek. Data Structures and Algorithms in C++ 4th Edition, Cengage Learning, (2012), ISBN: 978-1133608424

Other supplemental materials
- https://stepik.org/course/579/promo
https://dev.to/ayabouchiha/free-resources-to-master-algorithms-data-structure-2nfj

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Implement various data structures and their algorithms and apply them in implementing simple applications	C3 1, 10
CLO 2	Analyze simple algorithms and determine their complexities	C4 2, 6
CLO 3	Apply the knowledge of data structure to other application domains	C3 3,4, 8, 10
CLO 4	Design new data structures and algorithms to solve problems	C5 4,5, 8

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to Data structures
2	Abstract data types
3	Complexity analysis, Big Oh notation,
4	Arrays, traversing in a Linear Array, Searching in a Linear Array, Insertion in a Linear Array. Deletion in a Linear Array
5	linked list & its various types, sorted linked list,
6	searching an unsorted array, binary search for sorted arrays, hashing and indexing, open addressing and chaining,
7	Memory management and garbage collection.
8	Stacks (linked lists and array implementations), Recursion and analyzing recursive algorithms, divide and conquer algorithms,
9	Sorting algorithms (selection, insertion, merge, quick, bubble, heap, shell, radix, bucket),
10	queue, dequeuer, priority queues (linked and array implementations of queues),
11	trees and tree traversals, binary search trees, ,
12	Representation of Binary Tree, Using Linked List, Sequential Representation
13	Algorithms for Binary Tree Traversing(Preorder, Inorder, Post Order)
14	Insertion, Deletion and Searching operations with algorithms
15	Graphs, breadth-first and depth-first traversal, topological order, shortest path,
16	adjacency matrix and adjacency list implementations

CS212 - Database Systems

Credit Hours: 4(3+1) Prerequisites: None
Course Category: Computer Science Core Course

Description
The course aims to introduce basic database concepts, different data models, data storage and retrieval techniques and database design techniques. The course primarily focuses on relational data model and DBMS concepts. A brief introduction to NoSQL systems is also given.

Textbook(s)
- Thomas Connolly. Database Systems: A Practical Approach to Design, Implementation, and Management 6th Edition, Pearson, (2015), ISBN: 978-0132943260.

Other supplemental materials
- Jeffrey D. Ullman Hector Garcia-Molina, Jennifer Widom, Database Systems, Pearson, (2013), ISBN: 978-9332518674.

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Explain fundamental database concepts.	C2 1,2, 10
CLO 2	Design conceptual, logical and physical database schemas using different data models.	C5 1,2, 4
CLO 3	Identify functional dependencies and resolve database anomalies by normalizing database tables.	C2 3,4
CLO 4	Use Structured Query Language (SQL) for different operations in any DBMS	C3 4,5
CLO 5	Build different SQL databases	C5 4,5, 10

Week wise Lecture Schedule

W #	Lecture Topics
1	Basic database concepts, Database approach vs. file-based system
2	Database architecture, three level schema architecture, data independence
3	Relational data model, attributes, schemas, tuples, domains, relation instances,
4	Keys of relations, integrity constraints
5	Relational algebra, selection, projection, Cartesian product, types of joins
6	Entity relationship model, entity sets, attributes, relationship, entity-relationship diagrams
7	Normalization, functional dependencies, normal forms
8	Structured Query Language (SQL)
9	Data Query Language, Data Management Language
10	Data Definition Language, Data Control Language
11	Joins and sub-queries in SQL
12	Indexing, Triggers, Store Procedures
13	Transaction Management, ACID Properties
14	Concurrency, Database backup and recovery
15	NoSQL systems
16	NoSQL systems - MongoDB

CS213 - Software Engineering

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Computer Science Core Course

Description
This course provides a comprehensive introduction to software engineering, covering key topics such as professional software development, software process models, agile development techniques, requirements engineering, design and implementation, software testing and quality assurance, software evolution, project management, and configuration management. Students will gain a solid understanding of software engineering principles and practices, preparing them for successful software development projects.

Textbook(s)
- Sommerville I., Software Engineering, 10th Edition, Pearson Inc., (2017), ISBN: 978-9332582699

Other supplemental materials
- Roger Pressman, Bruce Maxim. ISE SOFTWARE ENGINEERING: A PRACTITIONERS APPROACH, McGraw-Hill Education, (2019), ISBN: 978-1260548006

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Describe various software engineering processes and activates	C1 1, 10
CLO 2	Apply the system modeling techniques to model medium size software systems	C3 2, 4, 5
CLO 3	Apply software quality assurance and testing principles to medium size software systems	C3 2,3
CLO 4	Discuss key principles and common methods for software project management such as scheduling, size estimation, cost estimation and risk analysis	C2 3,10

Week wise Lecture Schedule

W #	Lecture Topics
1	Nature of Software, The Unique nature of WebApp, Software Engineering, The software process, Software Engineering practice, software myth
2	Software Process Models: Generic and Prescriptive Models
3	Specialized Process Models, CMM, Process Assessment and Improvement
4	Agile Methods: Extreme Programming, Scrum, and Agile Principles
5	Unified Process Model, Personal and Team Process Models
6	Software Engineering Core principles, Understanding Requirements
7	Requirements modeling scenarios, information and analysis classes
8	UML diagrams
9	Data and Process Modeling Diagram, DFDs, Logical Vs Physical Models
10	Design Concepts: Principles, Component-level Design
11	Architectural Design: Styles, Patterns, and Case Studies
12	Quality concepts, Review Techniques
13	Software Testing Strategy and testing conventional Application
14	Software Configuration management
15	Introduction to project management, Project scheduling, Gantt Chart, Risk Management: Proactive VS Reactive Risk Management strategies
16	Maintenance, Reengineering, and Risk Management

CS214 – Design and Analysis of Algorithms

Credit Hours: 3(3+0) **Prerequisites:** CC211
Course Category: Computer Science Core Course

Description
Detailed study of the basic notions of the design of algorithms and the underlying data structures. Several measures of complexity are introduced. Emphasis on the structure, complexity, and efficiency of algorithms.

Textbook(s)
- Thomas H. Corman, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. Introduction to Algorithms, 3rd edition, MIT Press, (2009), ISBN: 978-0262033848
- Robert Sedgewick, Kevin Wayne. Algorithms, Fourth Edition, Deluxe Edition, (2015), ISBN: 978-0134384689

Other supplemental materials
- CS502- Fundamentals of Algorithms, Open Courseware, Virtual University of Pakistan
- https://www.coursera.org/learn/analysis-of-algorithms

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understanding Algorithms and Asymptotic Notations	C2 1,10
CLO 2	Apply algorithm Design Techniques	C3 2,3
CLO 3	Exploring Graph Algorithms and Complexity Classes	C4 4
CLO 4	List and contrast standard complexity classes	C4 4,6,10

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction; role of algorithms in computing
2	Analysis on nature of input and size of input Asymptotic notations; Big-O, Big Ω, Big Θ, little-o, little-ω
3	Recursion and recurrence relations Divide-and conquer technique: Merge Sort
4	Divide-and conquer technique: Quick Sort
5	Heaps, Heap Sort, Hashing
6	Algorithm Design Techniques, Brute Force Approach
7	Understand the concepts of Dynamic programming
8	Dynamic programming: Edit Distance
9	Dynamic programming: Chain Matrix Multiplication
10	Dynamic programming: Knapsack Problem
11	Greedy approach: Coin Change
12	Greedy approach: Huffman Encoding, Activity Selection
13	Graph algorithms, Graph traversing
14	Understand the concepts of MST and their algorithms: Kruskal's Algorithm, Prim's Algorithm
15	Understand the algorithms for computing shortest path: Dijkstra's Algorithm, Bellman-Ford Algorithm, Floyd-Warshall Algorithm
16	Introduction to complexity classes

CS215 - Operating Systems

Credit Hours: 3(3+0) **Prerequisites:** CS112
Course Category: Computer Science Core Course

Description
To help students gain a general understanding of the principles and concepts governing the functions of operating systems and acquaint students with the layered approach that makes design, implementation and operation of the complex OS possible.

Textbook(s)
- Abraham Silberschatz, Greg Gagne, Peter B. Galvin. Operating System Concepts 10th Edition, Wiley, (2021), ISBN: 978-1119800361
- Andrew S. Tanenbaum. Modern Operating Systems, 4th edition, Pearson, ISBN: 978-9332575776

Other supplemental materials
- Unix online Tutorial - https://www.tutorialspoint.com/unix/index.htm
- Operating Systems - https://www.tutorialspoint.com/operating_system/index.htm

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand the characteristics of different structures of the Operating Systems and identify the core functions of the Operating Systems	C2 1, 10
CLO 2	Analyze and evaluate the algorithms of the core functions of the Operating Systems and explain the major performance issues with regard to the core functions	C6 2, 8
CLO 3	Demonstrate the knowledge in applying system software and tools available in modern operating systems.	C3 3, 5

Week wise Lecture Schedule

W #	Lecture Topics
1	Operating systems basics
2	System calls, process concept and scheduling, inter-process communication
3	Multithreaded programming, multithreading models, threading issues
4	Process scheduling algorithms, thread scheduling, multiple-processor scheduling
5	Synchronization, critical section, synchronization hardware, synchronization problems
6	Deadlocks, detecting and recovering from deadlocks
7	Memory management, swapping, contiguous memory allocation, segmentation & paging
8	Virtual memory management, demand paging, thrashing, memory-mapped files
9	File systems, file concept, directory and disk structure, directory implementation
10	Free space management, disk structure and scheduling, swap space management,
11	System protection
12	Virtual machines
13	Operating system security
14	Introduction to Unix/Linux
15	Working in Bash Shell
16	Technical Review of Famous Operating systems like MS Windows, MacOS, Linux, Android etc.

CS311 - Computer Networks

Credit Hours: 3(2+1) **Prerequisites:** None
Course Category: Computer Science Core Course

Description
This course introduces the basic concept of computer network to the students. The network functionalities, procedures, protocols and various terminologies are discussed by taking the layered network models (OSI, TCP/IP). A simulation tool is also discussed. In the practical work the packet tracer is used for building different types of networks.

Textbook(s)
- James F. Kurose, Keith Ross. Computer Networking 8th Edition, Pearson, (2020), ISBN: 978-0136681557
- William Stallings. Data and Computer Communications, 10th Edition, Pearson, (2013), ISBN: 978-0133506488
- Behrouz A. Forouzan. Data Communication and Computer Networks, 5th Edition, McGraw Hill, (2012), ISBN: 978-0073376226

Other supplemental materials
- Packet Tracer Tutorial - https://www.packettracernetwork.com/tutorials/

- Online Networking Tools - https://manytools.org/network/

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Describe the key terminologies and technologies of computer networks	C2 1,2 ,10
CLO 2	Explain the services and functions provided by each layer in the Internet protocol stack.	C2 1,2,10
CLO 3	Analyze various internetworking devices and protocols and their functions.	C4 4,5
CLO 4	Build Computer Network on various Topologies	P3 4,5 ,6 ,8

Week wise Lecture Schedule

W #	Lecture Topics
1	Basic concepts of networking, Peer to peer and client server architectures,
2	Network topologies, LAN, MAN and WAN
3	Layered architecture, TCP/IP, OSI
4	Physical layer functionality, Broadcast vs. Point-to-Point Communication, Directional and Omni Directional Communication, Modes of Communication (Simplex, Half Duplex and Full Duplex).
5	Media: Bounded Media, Coaxial Cable, Twisted Pair (Shielded and Unshielded), Optical Fiber, Unbounded Media, Radio Waves, Microwave, Infrared, Satellites
6	Data link layer functionality, MAC addressing (unicast, multicast, broadcast) , CSMA, Types of Transmission (Asynchronous and Synchronous Transmission)
7	Multiplexing: Frequency Division Multiplexing, Time Division Multiplexing, Synchronous TDM & Statistical TDM
8	IEEE 802.x standards (Wired Networks)
9	Networking devices, Network layer protocols (IPv4 and IPv6)
10	Transport layer protocols, ports and sockets, TCP and UDP
11	Application layer protocols
12	Wireless Networks standards and technologies
13	Decentralized wireless networks
14	Cellular Networks, Generations and advancements
15	Latest trends in computer networks
16	Case studies

CS312 - Artificial Intelligence

Credit Hours: 3(2+1) **Prerequisites:** None
Course Category: Computer Science Core Course

Description
Artificial Intelligence has emerged as one of the most significant and promising areas of computing. This course focuses on the foundations of AI and its basic techniques like Symbolic manipulations, Pattern Matching, Knowledge Representation, Decision Making and Appreciating the differences between Knowledge, Data and Code. AI programming language Python has been proposed for the practical work of this course.

Textbook(s)
- Russell, S. and Norvig, P. Artificial Intelligence. A Modern Approach, 3rd ed, Prentice Hall, (2015), ISBN: 978-9332543515
- Philip Robbins. Python Programming for Beginners: The Complete Guide to Mastering Python, (2023), ISBN: 979-8376161821
- Kaplan, J. Artificial Intelligence: What Everyone Needs to Know 1st Edition, Oxford University Press, (2016), ISBN: 978-1493682225

Other supplemental materials
- Python Tutorial - https://www.learnpython.org

- Online Python Compiler - https://www.programiz.com/python-programming/online-compiler/

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand the fundamental constructs of Python programming language.	C2 5,10
CLO 2	Understand key concepts in the field of artificial intelligence	C2 1, 2,10
CLO 3	Implement artificial intelligence techniques and case studies	C3 3, 5, 6

Week wise Lecture Schedule

W #	Lecture Topics
1	An Introduction to Artificial Intelligence and its applications
2	AI history, branches of AI
3	Intelligent agents, PEAS
4	types of agents and architectures
5	Knowledge Based Systems, Introduction to Reasoning and Knowledge Representation
6	Problem Solving by Searching (Informed searching, Uninformed searching, Heuristics, Local searching)
7	Problem Solving by Searching (Min- max algorithm, Alpha beta pruning, Game-playing)
8	Case Studies: General Problem Solver, Eliza, Student, Macsyma
9	Rule based system, Channing method, Fuzzy Logic System
10	Machine Learning, Types (supervised and Unsupervised) Reinforcement Learning.
11	Learning from examples; ANN and ***Computer Vision***
12	More topics on Searching: Hill climbing, Simulated Annealing, Tabu Search, Min-Max Algorithm
13	Recent trends in AI and applications of AI algorithms
14	Introduction to Python programming language or LISP
15	Case study – 1 (e.g., AGI: J. Reed, et. Al, "A Generalist Agent", May 19, 2022)
16	Case study – 2

CS313 - Theory of Automata

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Computer Science Core Course

Description

This is an introductory course on Theory of Automata. Students are introduced to the concept of Formal Languages using Automata and Compiler. In Automata they learn about finite automata (deterministic; non-deterministic), transition graphs and pushdown automata (deterministic; non-deterministic). Introduces the foundations of formal language theory, computability, and complexity. This Course shows the relationship between automata and various classes of languages. Addresses the issue of which problems can be solved by computational means, and studies complexity of solutions

Textbook(s)

- Peter Linz Jones & Bartlett. Introduction to Formal Languages and Automata 6th Edition, Learning publisher (2016), ISBN: 978-1284077247
- Danial I.A Cohen John. Introduction to Computer Theory, Willy & Sons Inc (1986), ISBN: 0-471-80271-9
- John E. Hopcroft, Rajeev Motwani, Jeffery D. Ullman. Introduction to Automata Theory, language and Computation, 3rd Edition, Pearson Publisher (2008), ISBN: 9788131720479

Other supplemental materials

- https://www.tutorialspoint.com/automata_theory/index.htm
- https://www.javatpoint.com/automata-tutorial

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand the fundamental concepts of formal languages and automata theory, including terminologies, descriptive and recursive definitions, and regular expressions, to effectively analyze and manipulate languages.	C2 1,2
CLO 2	Demonstrate proficiency in designing and implementing Finite Automata (FA) and Deterministic Finite Automata (DFA) using transition tables, enabling accurate language recognition and processing.	C3 3,4
CLO 3	Develop a comprehensive understanding of advanced topics such as Non-Deterministic Finite Automata (NFA), conversion of NFA to DFA, and context-free grammars, enabling the analysis and construction of complex language structures.	C4 1,3

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction, Terminologies of Languages, Operations on languages
2	Descriptive definition
3	Recursive Definition
4	Regular Expressions (RE)
5	Transition Table, Finite Automaton (FA)
6	Deterministic Finite Automata- DFA
7	Martin Technique
8	Non-Deterministic Finite Automata- NFA
9	Conversion of NFA to DFA
10	Union of Two DFAs,
11	Concatenation of Two DFAs
12	Transition Graph-TG
13	Generalize Transition Graph (GTG)
14	Context Free Grammar-CFG
15	Total Language Tree
16	Turing machine

CS411 - Information Security

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Computer Science Core Course

Description
This course provides a broad overview of the threats to the security of information systems, the responsibilities and basic tools for information security, and the levels of training and expertise needed in organizations to reach and maintain a state of acceptable security. It covers concepts and applications of system and data security. Areas of particular focus include secure network design, implementation and transition issues, and techniques for responding to security breaches.

Textbook(s)
William Stallings, Lawrie Brown,. Computer Security: Principles and Practice 4th Edition, Pearson, (2017), ISBN: 978-0134794105
Michael E. Whitman, Herbert J. Mattord.Principles of Information Security 6th Edition, Cengage Learning, (2017), ISBN: 978-1337102063

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Explain key concepts of information security such as design principles, cryptography, risk management, and ethics	C2 1,10
CLO 2	Discuss legal, ethical, and professional issues in information security	A2 2,3,7 ,8,9
CLO 3	Apply various security and risk management tools for achieving information security and privacy	C3 4,5
CLO 4	Identify appropriate techniques to tackle and solve problems in the discipline of information security	C4 2,3,4

Week wise Lecture Schedule

W #	Lecture Topics
1	Information security Fundamentals: Define Security, Role & Significance, Applications, Beneficiaries, Characteristics of Information i.e. Confidentiality, Integrity, Availability, Authenticity, Utility etc, Terminologies of Information Security
2	Memory exploits, i.e. Buffer overflow (Stack Overflow, Heap Overflow, Integer Overflow) , Format String
3	Threats, its Types & Defenses. Virus, Malware, Spyware, Trojan, Phishing, Ransomware, Breaches etc.
4	Attacks, its Types & Defenses. Masquerade, Modification, Repudiation, Eavesdropping, Spoofing, DoS etc.
5	Crimes, its Types & Defenses. Crime, Computer Crime, Cyber Crime. Types of Cyber Crimes: Financial Fraud, Counterfeiting, ID Theft, Cyber Terrorism, Cyber Extortion, Cyber Warfare, Crypto Jacking, Cyber Bullying, Cyber Talking, Cyber Espionage, Cyber Squatting, Digital Piracy, IP Infringement, Privacy Invasion, Doxing, BEC & Exit Scam, Spam & Salami, Hacking, Botnet & Predator etc. Cyber Crimes Prevention and Combating Approaches
6	System Security: Defining System Security and its Objectives. Violations via Vulnerabilities, Threats, and Attacks. System Breaches via Confidentiality, Integrity, Authentication, ToS, and DoS
7	Information security Policies, Standard and practices . Types of security policies (Access control lists, Access control Metrix, Configuration Rules)
8	Access Control, Its types and Elements of Access Control . Access control Models (Mandatory access control, Nondiscretionary access control, Discretionary access control)
9	Network Security
10	Cryptography, Definitions and Concepts,
11	Cipher Methods (Bit Stream, Block Stream). Substitution Cipher (Monoalphabetic substitution, Polyalphabetic substitution, Vigenère cipher)
12	Transposition Cipher (Caeser Matrix, Exclusive OR (XOR), Varnam Cipher, Book or Running Key Cipher)
13	Symmetric Key Cryptography. Private Key Distribution Approaches. Stream Ciphers vs. Block Ciphers. DES Algorithm vs. AES Algorithm
14	Asymmetric Key Cryptography. Public Key Distribution Approaches. RSA Algorithm vs. PGP Application. Comparison of Symmetric and Asymmetric Key Cryptography
15	Intrusion Detection and Prevention Systems
16	Signature based IDPS, Behavior Based IDPS, Network-based IDPS, Host-based IDPS

CS412 - Computer Organization and Architecture

Credit Hours: 3(3+0) **Prerequisites:** CS113
Course Category: Computer Science Core Course

Description

The course in Computer Architecture explores the fundamental principles that govern the design and functioning of modern computer systems. Students will explore the evolution of key architectures, performance considerations, interconnection structures, memory systems, and processor organization. This course provides a comprehensive understanding of the underlying mechanisms that contribute to the efficiency and functionality of computer systems.

Textbook(s)

- Computer Organization and Architecture 10th Edition by William Stallings,(2016), ISBN: 978-0134101613.

Other supplemental materials

- Introduction to Computer Organization: An Under the Hood Look at Hardware and x86-64 Assembly by Robert Plantz, (2022), ISBN: 978-1718500099.
- Kip Irvine. Assembly Language for Intel Based Computers, 5th edition, Prentice Hall, (2007), ISBN: 978-0132383103.

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Develop a foundational understanding of computer architecture basics.	C4 1, 2, 10
CLO 2	Examine principles for designing high-performance computer systems, including considerations for multicore processors, MICs, and GPGPUs.	C5 1, 6
CLO 3	Acquire the skills to measure and evaluate computer performance, utilizing fundamental metrics and performance indicators to gain insights into the efficiency and functionality of computer systems.	C2 1, 2, 4,10
CLO 4	Understand the top-level view of computer function and interconnection.	C1 1, 2,4
CLO 5	Understand the memory hierarchy, covering cache memory principles, semiconductor main memory, and external memory options. Gain insights into processor structure, organization, registers, and the instruction cycle, forming a holistic understanding of computer architecture and organization.	C1 1, 2,3, 10

Week wise Lecture Schedule

W #	Lecture Topics
1	Basics: Evolution of Intel x86 Architecture, Embedded Systems, ARM Architecture, Introduction to Cloud Computing
2	Performance Issues: Designing for Performance, Multicore Architectures, MICs (Many Integrated Cores), and GPGPUs
3	Performance Metrics: Amdahl's Law and Little's Law, Basic Measures of Computer Performance
4	Introduction to Computer Architecture: Top-Level View of Computer Function, Main Components, and Functional Overview
5	Interconnection Structures: Bus Interconnection, System Bus Structure, Control and Timing, PCI Express
6	Point-to-Point Interconnects: Comparison with Bus Systems, High-Speed Interconnect Technologies
7	Computer Memory System Overview: Memory Hierarchy, Characteristics of Memory Systems
8	Cache Memory Principles: Cache Mapping Techniques, Replacement Policies, Write Policies
9	Elements of Cache Design: Associativity, Block Size, Hit Rate, and Latency
10	Pentium 4 Cache Organization: L1/L2 Cache, Cache Coherence, Performance Considerations
11	Internal Memory: Types of Semiconductor Memory, SRAM vs DRAM, DDR DRAM, ECC (Error Correction Code)
12	External Memory: Overview of Magnetic Disks, Disk Performance Parameters
13	RAID Concepts: RAID Levels, Performance and Reliability Tradeoffs
14	Solid State Drives (SSD): Architecture, Performance Benefits, Limitations; Optical Memory Basics
15	Processor Structure: Basic Processor Components, Control Unit, ALU, Datapath
16	Instruction Cycle and Execution: Register Organization, Instruction Fetch and Decode, Micro-operations

Specialization – General Computer Science (Honors)

CS321 - Web Technologies

Credit Hours: 3(2+1) **Prerequisites:** None
Course Category: Sp – Gen. Computer Science

Description

This course provides an introduction to the fundamentals of web development. Students will learn about front-end and back-end development basics, including HTML, CSS, JavaScript, and server-side programming. They will acquire the skills to create interactive and user-friendly web interfaces. The course also covers database integration, full-stack development, content management systems, deployment and hosting considerations, as well as version control and collaboration practices. By the end of the course, students will have a solid understanding of web development principles and the ability to create functional web applications.

Textbook(s)

- Riaz Ahmed. Full Stack Web Development for Beginners: Learn Ecommerce Web Development Using HTML5, CSS3, Bootstrap, JavaScript, MySQL, and PHP, (2021), ISBN: 979-8738951268
- Frank Zammetti. Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, Python, Django, and Docker 2nd ed. Edition (2020), ISBN: 978-1484257371

Other supplemental materials

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Demonstrate an understanding of the fundamentals of web development, including front-end and back-end development basics	C2 1,2, 10
CLO 2	Apply front-end development skills to create interactive and user-friendly web interfaces	C3 2, 4, 5,6
CLO 3	Understand the process of deploying and hosting web applications, including version control and collaboration	C2 3, 5, 6, 10

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to Web Development: Overview of web technologies, Introduction to HTML and CSS, Setting up a development environment
2	Front-End Development Basics: HTML markup and structure, HTML simple tags
3	Front-End Development Basics: HTML advance tangs
4	Front-End Development Basics: CSS styling and layout
5	Front-End Development Basics: Responsive design, CSS frameworks (e.g., Bootstrap)
6	Front-End Development Basics: JavaScript syntax and data types, Working with variables, arrays, and objects
7	Front-End Development Basics: Applying JQuery, handing actions, DOM manipulation and event handling
8	Back-End Development Basics: Introduction to server-side programming languages (e.g., Node.js, Python, PHP), Installing a webserver and understanding how a page request is processed, Understanding the language syntax – variables, arrays, functions, Handling HTTP requests and responses
9	Database Integration: Introduction to databases (e.g., MySQL, MongoDB), Working with databases and SQL, querying databases and retrieving data
10	Full-Stack Development: Combining front-end, back-end components, and database components, submitting form data to a database, retrieving data from database, and displaying on a webpage
11	Full-Stack Development: Performing CRUD operations, Deployment and hosting considerations
12	Full-Stack Development: Building a complete web application, A complete study to understand form submission, retrieval data, display data, edit form, giving a single page/form multi-view
13	A complete case study from front-end design to dynamic server-side functionality
14	An introduction to Content Management Systems: E.g., WordPress, Joomla, Shopify, Drupal
15	Deployment and hosting considerations
16	Version Control and Collaboration: Introduction to version control systems (e.g., Git) Managing code repositories and branching, Collaborative development workflows Portfolio Development

CS322 - Advanced Programming

Credit Hours: 3(2+1) **Prerequisites:** None
Course Category: Sp – Gen. Computer Science

Description

This advanced programming course typically focuses on advanced programming concepts, techniques, and best practices in high level language C#.NET which will help students enhance their programming skills and build complex software applications for desktop, web and games. They will learn how to write maintainable, scalable, and readable code.

Textbook(s)

- Joyce Farrell. Microsoft Visual C# 2015: An Introduction to Object-Oriented Programming 6th Edition, Cengage Learning Publisher (2015), ISBN: 978-1285860237
- Gabriel Baptista, Francesco Abbruzzese. Software Architecture with C# 10 and .NET 6 Develop software solutions using microservices, DevOps, EF Core, and design patterns for Azure, Packt Publishing 3rd Edition,(2022), ISBN: 978-1803235257
- Jon Skeet. C# in Depth, Manning Publisher, Fourth Edition, (2019), ISBN: 978-1617294532

Other supplemental materials

Online Tutorial - https://www.w3schools.com/cs/index.php
Online Compiler - https://www.programiz.com/csharp-programming/online-compiler/

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Describe and explain the functional mechanisms of Windows platform	C2 1, 10
CLO 2	Identify the objects & their relationships to build object-oriented solution	C3 2, 6
CLO 3	Model a solution for a given problem using object-oriented principles	C3 3,4, 8, 10
CLO 4	Examine an object-oriented solution	C4 5,6, 8

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to Visual Studio .NET IDE setup for C# development.
2	Memory concepts, operators, control structures, loops, arrays (including jagged), foreach loop in C#.
3	OOP in C#: namespaces, classes, access levels, static vs instance, abstract classes, assemblies, exception handling.
4	Windows Forms UI: graphical controls, event handling, delegates, MDI, menus, and advanced controls.
5	ADO.NET database connectivity: key classes, SQL commands, data binding, GridView, Cloud SQL access.
6	Files and Streams: file/directory handling, static classes, FileStream usage with examples.
7	Multithreading: thread lifecycle, priorities, synchronization (Mutex, Monitor, Semaphore), threading classes.
8	ASP.NET basics: Web Forms, MVC, IIS, Web Services, page lifecycle, event model, code-behind.
9	ASP.NET HTTP: request/response, HttpRequest/HttpResponse classes, redirection, HttpUtility.
10	ADO.NET & LINQ: data providers, connected/disconnected access, data controls, data binding.
11	Data source controls: GridView, DetailsView, ListView, Entity Data Model, ASP.NET AJAX essentials.
12	Xamarin intro: cross-platform mobile apps, Xamarin.Forms, UI with XAML and code.
13	Xamarin.Forms UI controls, layouts, images, interactive lists, navigation patterns.
14	Xamarin.Forms: forms, settings, local storage (files, SQLite), RESTful services, MVVM architecture.
15	Case studies, Projects
16	Projects and Presentations

CS323 - Web Engineering

Credit Hours: 3(2+1) **Prerequisites:** CS371
Course Category: Sp – Gen. Computer Science

Description

Expand your web development skills with this advanced course. Discover advanced techniques in HTML, CSS, and JavaScript, enabling you to create visually stunning and interactive web pages. Explore popular frameworks like Laravel, Node.js, and React to build robust full-stack applications. By the end of this course, you'll have the expertise to develop professional-grade web applications and take your web development skills to new heights.

Textbook(s)

- Christopher John Pecoraro. Mastering Laravel, Packt Publishing, (2015), ISSN: 978-1785285028
- Matt Stauffer. Laravel: Up & Running: A Framework for Building Modern PHP Apps, 2nd Edition, O'Reilly Media, (2019), ISSN: 978-1492041214
- Sebastian Springer. Node.js: The Comprehensive Guide to Server-Side JavaScript Programming, Rheinwerk Computing, (2022), ISSN: 978-1493222926

Other supplemental materials

https://laracasts.com/series/laravel-8-from-scratch
https://laracasts.com/topics/react

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Demonstrate proficiency in advanced HTML and CSS techniques	C3 1,2, 10
CLO 2	Apply advanced JavaScript concepts to develop interactive and dynamic web applications	C4 3, 4
CLO 3	Build full-stack applications using modern frameworks such as Laravel, Node.js, and React	C5 2, 5, 6, 10

Week wise Lecture Schedule

W #	Lecture Topics
1	Advanced HTML and CSS Techniques: HTML5 advanced features and APIs. CSS preprocessors (e.g., SASS, LESS), Responsive design and media queries
2	Advanced JavaScript Concepts: Advanced JavaScript syntax and features, Promises and async/await, Manipulating the DOM with advanced techniques
3	Introduction to Laravel Framework: Overview of Laravel and its ecosystem, Setting up a Laravel project, Initial environment setup and composer
4	Laravel Basics: Understanding MVC architecture, understanding various file/folders in Laravel, routing requests, and linking
5	Laravel Basics: Including CSS and Javascript in Laravel, Blade Layout
6	Laravel Database Integration: Working with databases using Laravel's ORM (Eloquent), Database migrations and seeding
7	Advanced Laravel Features: Authentication and authorization in Laravel, Caching and performance optimization techniques, Building RESTful APIs with Laravel
8	Introduction to Node.js: Overview of Node.js and its architecture, Asynchronous programming with Node.js, Building a basic Node.js server
9	Express.js Framework: Introduction to Express.js framework, Routing and middleware in Express.js, Handle different types of requests (GET, POST, PUT, DELETE, etc.)
10	Express.js Framework: Integrating databases with Node.js using ORMs (e.g., Sequelize)
11	Advanced Node.js Development: Implementing authentication and authorization in Node.js
12	Introduction to React: Understanding the React component model, JSX syntax and rendering elements
13	Introduction to React: State management and props in React
14	React Components and Hooks: Building reusable components in React, React hooks for state and side-effects
15	React Components and Hooks: React router for client-side routing
16	Building Full-Stack Applications: Integrating React and Node.js (MERN stack). Creating RESTful APIs with Node.js and Express, building a complete web application using the MERN stack

CS421 - Machine Learning

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp – Gen. Computer Science

Description
Machine learning is one of the fastest growing areas of computer science, with far-reaching applications. The aim of this course is to: a) Present the basic machine learning concepts; b) Present a range of machine learning algorithms along with their strengths and weaknesses; c) Apply machine learning algorithms to solve problems of moderate complexity.

Textbook(s)
- Alpaydin, E. Introduction to Machine Learning. Cambridge, MA: The MIT Press. (2020), ISBN: 978-0262043793
- Bishop, C. M. Pattern Recognition and Machine Learning. New York, NY: Springer. (2016), ISBN: 978-1493938438

Other supplemental materials
- Kaggle - https://www.kaggle.com
- Google colab
https://colab.research.google.com/?utm_source=scs-index

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Describe basic machine learning concepts, theories and applications.	C1 1,10
CLO 2	Apply supervised learning techniques to solve classification problems of moderate complexity.	C3 6,7,8, 9,10
CLO 3	Apply unsupervised learning techniques to solve clustering problems of moderate complexity	C3 2,3, 4, 5
CLO 4	Apply reinforcement learning algorithms to environments with complex dynamics.	C3 1,2, 3
CLO 5	Develop a reasonable size project using suitable machine learning technique	C6 1,2, 3,4

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to Machine Learning, Types of learning (supervised, unsupervised, semi-supervised, reinforcement), Real-world applications
2	Concept learning, General-to-specific ordering of hypotheses, Hypothesis spaces
3	Version Spaces Algorithm, Candidate Elimination Algorithm, Inductive bias
4	Introduction to Supervised Learning, Decision trees, ID3 algorithm, Splitting criteria
5	Naive Bayes classifier, Assumptions, Training and prediction, Use cases
6	Artificial Neural Networks, Perceptron learning, Activation functions
7	Support Vector Machines (SVM), Linear and non-linear classification, Kernels
8	Overfitting and underfitting, Dealing with noisy data, Pruning techniques
9	Measuring classifier performance, Confusion matrix, Precision, Recall, F1-score, ROC curve
10	Linear Regression, Least squares method, Cost functions
11	Logistic Regression, Sigmoid function, Gradient descent for optimization
12	Random Forest,
13	Ada-boost, XG Boost,
14	Unsupervised Learning, Hierarchical Agglomerative Clustering, Dendrograms
15	Case Study/Project/Demo
16	Presentation/Peer Review/Q&A Sessions/Demo

CS422 - Mobile Application Development

Credit Hours: 3(2+1) **Prerequisites:** None
Course Category: Sp – Gen. Computer Science

Description

Mobile app development creates software for smartphones and tablets, mainly on Android and iOS. It's rapidly growing as industries—from retail to healthcare—need real-time, convenient access for users. Mobile apps are now the primary way people and businesses connect online. To stay competitive, organizations must meet these demands despite device limitations and distribution challenges. Following best practices can simplify this development process.

Textbook(s)

- Bill Phillips and Brian Hardy. Android Programming: The Big Nerd Ranch Guide, Addison-Wesley Professional, (2013), ISBN: 978-0321804334
- Eric Windmill, Flutter in Action, Manning, ISBN: 978-1617296147
- Dawn Griffiths, David Griffiths.Head First Android Development: A Learner's Guide to Building Android Apps with Kotlin 3rd Edition, O'Reilly Media ,(2021), ISBN: 978-1492076520

Other supplemental materials

Android Developer Documentation - https://developer.android.com
Flutter Documentation - https://docs.flutter.dev

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Describe Mobile Application Development fundamentals and flow on multiple devices and publishing it online	C2 1,3,4, 5,10
CLO 2	Develop Mobile Application using provided assets with basic functionality	C5 2,3, 6, 10
CLO 3	Generate Mobile application that uses hardware and software resources like sensors and configuration etc. and evaluate functionality	C5 1,3, 5, 7,10
CLO 4	Deploy mobile applications to relevant app stores (e.g., Google Play Store, Apple App Store)	C3 *2,4,5 ,10*

Week wise Lecture Schedule

W #	Lecture Topics
1	Overview of Android platform and its architecture. Setting up the development environment (Android Studio, SDK installation). Introduction to Java and Kotlin programming languages for Android development.
2	Understanding Android layouts and views Creating interactive UI elements (buttons, text fields, etc.)
3	Handling user input and events Designing responsive and adaptive layouts
4	Working with SQLite database for local data storage. Using Shared Preferences for simple data persistence.
5	Fetching and storing data from/to web services (RESTful APIs) Managing data using Content Providers
6	Making HTTP requests using Retrofit or Volley libraries Parsing and processing JSON/XML responses
7	Handling network connectivity and error scenarios Authentication and security considerations
8	Integrating multimedia elements (images, audio, video) Accessing device features like camera, sensors, and location
9	Utilizing permissions and requesting user consent Working with notifications and background services
10	Preparing an app for deployment (debugging, testing) Publishing an app on the Google Play Store
11	App store guidelines and submission process Monetization strategies (in-app purchases, ads, subscriptions)
12	Advanced UI techniques (custom views, animations) Performance optimization and memory management
13	Testing and debugging Android applications Exploring emerging trends in Android development (e.g., Jetpack Compose)
14	Applying the concepts learned throughout the course to develop a complete Android application
15	Iterative development, testing, and refinement of the project
16	Demonstrate critical thinking and problem-solving skills when facing challenges and complexities in mobile app development projects.

CS423 - Internet of Things and Its Applications

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp – Gen. Computer Science

Description
This course provides a comprehensive introduction to the basic concepts of the Internet of Things (IoT) and explores its wide range of applications. Students will learn the basic concepts, principles, and technologies behind IoT systems, as well as their impact on various industries. The course will cover essential topics such as sensors and actuators, connectivity protocols, data analytics, security, and privacy considerations. Students will also receive practical experience through practical exercises and projects involving IoT devices and platforms.

Textbook(s)
- Rajkumar Buyya, Amir Vahid Dastjerdi, and Anton Y. Dong . Internet of Things: Principles and Paradigms, Morgan Kaufmann Piublishers, (2016), ISBN: 978-0128053959
- Pethuru Raj and Anupama C. Raman. The Internet of Things: Enabling Technologies, Platforms, and Use Cases, CRC Press, (2017), ISBN: 978-1032339719

Other supplemental materials
- Research papers on IoT
- https://theinternetofthings.eu/
-https://www.techtarget.com/iotagenda/

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand the fundamental concepts, principles, and architectures of the Internet of Things.	C2 *1, 10*
CLO 2	Identify and evaluate different types of sensors, actuators, and connectivity protocols used in IoT systems.	C2 *2,3*
CLO 3	Examine real-world applications and case studies of IoT across different industries.	C3 *3,4,10*

Week wise Lecture Schedule

W #	Lecture Topics
1	Internet of Things Definition and characteristics, Key concepts of IoT systems
2	IoT architecture
3	IoT application in different domains
4	Enabling Technologies in IoT, History of IoT, About Things in IoT
5	IoT frameworks, IoT and M2M.
6	Sensors: Types of sensors and their characteristics
7	Actuators and their functionalities
8	Interfacing sensors and actuators with IoT platforms
9	Wireless communication protocols (e.g., Wi-Fi, Bluetooth, Zigbee)
10	IoT device connectivity options (e.g., RFID, NFC)
11	Edge computing and fog computing in IoT
12	IoT security challenges and threats Authentication and access control in IoT
13	Smart homes and buildings Industrial IoT and smart manufacturing (research papers)
14	Healthcare and wearable devices Smart cities and transportation (research papers)
15	Agriculture and environmental monitoring
16	Analysis of real-world IoT case studies Presentations and discussions on IoT projects

CS424 - Network Security

Credit Hours: 3(3+0) **Prerequisites:** CC261
Course Category: Sp – Gen. Computer Science

Description
This course is intended to provide the students not only a broad coverage of the typical network security topics, but also a practical and real-world exposure to network security challenges faced by professionals in the workplace.

Textbook(s)
- Jie Wang, Zachary A. Kissel. Introduction to Network Security: Theory and Practice, , John Wiley & Sons, (2015) ISBN: 1118939484, 9781118939482
- Shafiullah Khan, Al-Sakib Khan Pathan. Wireless Networks and Security, Springer Berlin Heidelberg, (2013), ISBN: 978-364236168-5.

Other supplemental materials
- Nitesh Dhanjani, Justin Clarke. Network Security Tools, , O'Reilly Media, Inc, (2005), ISBN: 9780596007942.

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand Network Security Fundamentals	C2 *1, 10*
CLO 2	Apply Cryptographic Techniques	C3 *2, 4, 5*
CLO 3	Analyze Key Distribution and User Authentication	C4 *4, 6*
CLO 4	Assess Network Security Protocols and Technologies	C6 *3, 10*

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to network security concepts.
2	Attacks and types. Security requirements
3	Secured network architecture, OSI Layer wise security
4	Physical layer security attacks and countermeasures
5	MAC layer security attacks
6	MAC layer security countermeasures
7	Network layer security attacks
8	Network layer security countermeasures
9	Transport layer security attacks and countermeasures
10	Application layer security attacks and countermeasures
11	Key management schemes and digital signature
12	AAA server
13	Wireless Network Security, IEEE 802.11 Wireless LAN, IEEE 802.11i Wireless LAN Security
14	Decentralized networks security
15	Intrusion Detection Systems and types
16	Case study - SNORT

CS425 – Digital Image Processing [TODO]

Credit Hours: 3(3+0) **Prerequisites:** CC261
Course Category: Sp – Gen. Computer Science

Specialization – Software Engineering

CS331- Software Requirement Engineering

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Software Engineering

Description

"Software Requirements Engineering" Is the first course in our "Software Engineering" series. If you are involved in software projects, especially large ones, you know that dealing with software requirements is the biggest problem the software practitioner faces almost at every project. Not working effectively with software requirements could be the number one reason for many software projects' failures. The course will discuss concepts for systematically establishing, defining, and managing the requirements for large, complex, changing, and software-intensive systems from technical, organizational, and management perspectives. The course will consider the past, present, and future software requirements effectively and efficiently. In this course, you will learn various techniques to capture requirements, analyse and validate requirements, control, and manage requirements change requests, and deliver a solid requirements document. You will understand the requirements analysts' role in a software project, which will help you know what you need from them and what they need from you.

Textbook(s)

- Karl Wiegers. Software Requirements , Microsoft Press ,(2003) ISBN: 978-0735618794
- Roger Pressman "Software Engineering: A Practitioner's Approach" McGraw Hill publisher, (2019), ISBN: 978-1260548006

Other supplemental materials

- Elizabeth Hull, Ken Jackson and Jeremy Dick. "Requirements Engineering" 4th edition, Springer-Verlag London Limited, (2017), ISBN: 978-3319869971

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand of the importance of following a systematic requirement engineering process	C2 *1, 4, 10*
CLO 2	Effectively gather and analyse software requirements for the development of cost-effective and efficient technical solutions	C3 *6,7, 8, 9, 10*
CLO 3	Use system modelling techniques for requirements analysis and requirements presentation	C4 2,3 ,5, 10

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to software Requirement Engineering, Software Requirements, Types of Software Requirements
2	Another view of requirements, Processes and Process Models, Requirements Engineering Process
3	Actors and stakeholders in the requirements engineering process, Social and Cultural Issues in Requirements Engineering
4	Requirements Elicitation, Requirements Elicitation Techniques
5	Requirements Analysis, Requirements Analysis and Negotiation
6	Requirements Errors, types of Requirements Errors,
7	Requirements Validation, Requirement Validation Techniques
8	Requirements Management, Requirements Traceability
9	Prototyping, Writing Requirements, Requirements Document
10	Quality Attributes of Requirements Documents, Use Case Modeling, Case Study and Use Cases for Case Study
11	Software Modeling, Object-Oriented Modeling with UML
12	Object-Oriented Static Modeling of the Banking System
13	Functional Modelling, Real-Time Structured Analysis and Design Technique (RSTAD) and Application of Banking Case
14	Structured Analysis and Design Technique, Dynamic Modelling, Dynamic Modelling of Banking System Case Study - I
15	Dynamic Modelling of Banking System Case Study – II, Requirements Document for the Banking System, Requirements Engineering for Agile Methods
16	Requirements Engineering for Web-Based Information Systems, Requirements Engineering for Product Lines, Requirements Prioritization

CS332 - Software Design and Architecture

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Software Engineering

Description
Introduction to OO Design, System design and software architecture, Object design, Mapping design to code, User interface design, Persistent layer design, Web applications design, State machine diagrams and modeling, Agile software engineering, Exploring inheritance, Interactive systems with MVC architecture, Software reuse.

Textbook(s)
- Len Paul Bass, Clements, Rick Kazman, Software Architecture in Practice, Addison-Wesley Professional, (2021), ISBN: 978-0136886099
- Moises Gamio. Software Design Principles: A Practical Guide, (2022), ISBN: 979-8848053401

Other supplemental materials
- Ludwin Barbin. SOFTWARE DESIGN PRINCIPLES Kindle Edition by, (2013), B00EHWS180
https://www.gofpatterns.com/index.php

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand the role of design and its major activities within the OO software development process, with focus on the Unified process	C2 1, 4, 10
CLO 2	Design OOD models and refine them to reflect implementation details	C5 3,4, 5
CLO 3	Implement the design model using an object-oriented programming language	C5 2,3, 5

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to Software Design: Introduction to Software Architecture, Introduction to Software Design, Design Vs. Architecture
2	Software Architecture Core Concepts: Define Software Architecture, Architecture Address NFR, Architectures and Technologies
3	Software Architecture Business Cycle: Where do architecture comes from, Software process and architecture business cycle, what makes a good architecture
4	Software Architecture Business Cycle: Activities in software process and architecture business cycle, Architectural Structures and views
5	Quality Attributes in Software Architecture: Motivation, what are the different Quantity attributes, Performance, Scalability, Modifiability, Security, Availability, Robustness, How to Achieve quality attributes at architectural level
6	Software Architecture Process: Process Outline, Architecture Design, Validation
7	Case Study (solution): Architecture Patterns, Structural View, Behavioral View, Implementation Issues
8	Documenting Software Architecture: Architecture Requirements, Solution, Architecture Analysis
9	Introduction to design: Introduction to Unified modelling language (UML), Structural and behavioral diagrams
10	Introduction to design: Introduction to Unified modelling language (UML), Structural and behavioral diagrams
11	Activity: Modeling structural (class diagram) and behavioral diagram (use case, sequence diagram and state machine diagram) of online shopping application
12	Introduction to Software Product Line engineering: Overview, what makes SPL works, Architectures for Product Lines
13	Aspect Oriented Architecture: Introduction to Aspect-Oriented Programming (basic concepts), Aspect oriented architecture
14	Aspect Oriented Architecture: Aspect oriented Modelling using UML, Aspect Oriented Modelling tools
15	Model-Driven Architecture: What is MDA? Why MDA?
16	Model-Driven Architecture: State of the Art Practices and Tools

CS333 - Human Computer Interaction

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Software Engineering

Description

Human-Computer Interaction (HCI) is the study of the principles and methods with which one builds effective interfaces for users. A basic precept of HCI is that users should be able to focus on solving problems, rather than dealing with the intricacies of complex software. Interfaces must be accessible, meaningful, visually consistent, comprehensive, accurate, and oriented around the tasks that users tend to perform. The course will provide a balance of practical and theoretical knowledge, giving students experience ordinarily not provided by other courses in computer science.

Textbook(s)

- Gerard Jounghyun Kim. Human-Computer Interaction: Fundamentals and Practice, , Auerbach Publications, (2015), ISBN: 9781482233896
- Helen Sharp, Jennifer Preece, Yvonne Rogers. Interaction Design: Beyond Human-Computer Interaction, Wiley; 5th edition (2019), ISBN-13 : 978-1119547259
- Alan Dix.Janet Finlay. Human Computer Interaction, 3rd Ed, (2016) ISBN: 9780130461094.

Other supplemental materials

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Acquire fundamental concepts of computer components functions regarding interaction with human and vice versa.	C2 *1,10*
CLO 2	Analyze interface problems to recognize what design approach and interaction styles is required in the light of usability standards.	C4 *2,3,10*
CLO 3	Apply basic concepts to construct a user-interaction strategy for a given problem its usability evaluation and	C3 *2,3,4, 10*
CLO 4	Ability to design and develop an interface by using appropriate HCI techniques that are preferred by the user.	C5 *5, 6,10*

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to HCI, The Goals of HCI Usability, HCI & its Evolution. Input/Output Channel
2	Human Senses and Characteristics. Eyesight, Hearing, Touch, Human Memory, Emotions etc.
3	The Computer, Text Entry Devices
4	Positioning, Pointing & Drawing Devices, Display, Devices
5	Physical Controls, sensors & special devices
6	The Interaction, What are Goals, Domain, Intention. Task Analysis, User& Task Language, and Intention
7	Models of Interaction, The Execution-evaluation cycle, The interaction
8	Frame Work, Ergonomics, Interaction Styles, Command line, Menus, WIMP, Natural Language, Point & Click, and Forms Fill-Ins etc
9	Elements of WIMP, Paradigm & Interaction Design Basics, Paradigm Introduction, Paradigm for Interaction, Time Sharing, VDUs
10	Programming Toolkit, Personal Language Vs Actions, Hypertexts, Multimodality, Computing, Direct Manipulations
11	Computer supported Cooperative Works, The WWW, Agent Based Interfaces, Ubiquitous Computing Sensor Based and Context aware Interaction, Interaction Design Basics, The Process of design, Design Goals Constraints and Trade-Offs, Golden Rule of Design, User Focus, Scenario
12	Navigation Design, Screen design and layout Iteration & Prototype, Design Rules, Principles to support usability, Standards
13	Guidelines, Golden Rules & Heuristics, HCI Patterns
14	Evaluation Techniques, What is Evaluation? Goals of Evaluation, Evaluation through expert analysis
15	Evaluation through user participation, Observational Techniques, Choosing an Evaluation method
16	Universal Design, Universal Design Principles, Multi-Modal Interaction, Design for diversity

CS431 – Formal Method and Design
[todo]

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Software Engineering

By the end of this course the student will be able to:
• Explain the role of formal methods in software specification and verification. (C2:
Understand)
• Apply formal mathematical methods and tools to solve elementary computation
problems. (C3: Apply)
• Use mathematics based techniques and tools for specifying, designing, and verifying
software. (C3: Apply)

CS433 - Software Project Management

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Software Engineering

Description

This course equips learners with essential project management principles and techniques for IT projects. Participants will critically evaluate real-world project management issues and collaborate in teams to develop comprehensive project plans. Through practical examples, learners will gain confidence in managing projects effectively.

Textbook(s)

- Bob Hughes and Mike Cotterell. Software Project Management, McGraw-Hill Education; 5th Edition (2009). ISBN-10: 0077122798

Other supplemental materials

- Project management Institute. A Guide to the Project Management Body of Knowledge, 3rd Edition (PMBOK Guides), (2021) ISBN-13: 978-1930699458

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Demonstrate an understanding of project lifecycle principles and the ability to identify suitable project scenarios to share this knowledge with learners effectively.	C2 *1, 10*
CLO 2	Engage in critical discussions with course participants and learners to assess the challenges and practical applications of project management in real-world settings	C6 *2, 3, 10*
CLO 3	Apply project management techniques for IT projects, including initiation, planning, execution, and evaluation, while collaborating in teams to develop a comprehensive project plan. This plan should encompass key tasks, critical path analysis, task dependencies, and a realistic timeline.	C3 *3, 4, 5*
CLO 4	Present practical strategies for building confidence in project management by illustrating simple project planning examples.	C5 *4,5,6*

Week wise Lecture Schedule

W #	Lecture Topics
1	**Introduction to Software Project Management**: Importance in IT, fundamentals of People, Process, Product, and Technology.
2	**Project Planning & Structures**: Planning, tracking, measurement; organizational structures and their influence.
3	**Initiating a Project**: Project Charter, Statement of Work (SOW), Request for Proposal (RFP).
4	**Planning Phase**: Development lifecycle models, creating project plans.
5	**Project Selection & Financial Analysis**: NPV, ROI, Payback, Weighted Scoring Model.
6	**Estimation & Budgeting**: Effort and cost estimation, developing project budgets.
7	**Scheduling**: Network diagrams, PERT, Gantt charts, Critical Chain Scheduling.
8	**Risk & Change Management**: Risk analysis/mitigation, change control processes.
9	**Development Management**: Team models, requirements process, configuration management.
10	**Metrics, Tools, & Team Management**: Software metrics, programming tools, conflict management, motivation.
11	**Project Control**: Status reporting, project metrics, Earned Value Analysis (EVA).
12	**Communication & Process Improvement**: Communication techniques, identifying process improvements.
13	**MS Project**: Resource leveling, various MS Project views.
14	**System Testing**: Test specifications, black-box and white-box testing.
15	**Final Phases**: Project recovery, documentation, cutover/migration, post-project reviews.
16	**Project Success Factors**: Management support, expectation handling, success metrics.

CS434 - Parallel and Distributed Computing

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Software Engineering

Description

"Parallel and Distributed Computing" is an intensive course covering essential topics in high-performance computing. Explore concepts such as concurrency control, fault tolerance, GPU programming, interconnection topologies, and load balancing. Gain practical experience with tools like Cuda, OpenMP, Hadoop, and more. Develop expertise in parallel algorithms, performance analysis, and scalability. By the end, you'll be equipped to design and optimize efficient computing systems for tackling complex computational challenges.

Textbook(s)

- Maarten van Steen, Andrew S. Tanenbaum . Distributed Systems, (2023) ISBN: 978-9081540636

- Ajit Singh. Parallel And Distributed Computing Kindle Edition, (2021), B09DMP9JT4

Other supplemental materials

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Learn about parallel and distributed computers.	C2 1,2, 10
CLO 2	Write portable programs for parallel or distributed architectures using Message-Passing Interface (MPI) library.	C3 2,3, 4
CLO 3	Analyze complex problems with shared memory programming with openMP.	C4 4,5, 6,10

Week wise Lecture Schedule

W #	Lecture Topics
1	Asynchronous/synchronous computation/communication
2	Concurrency control, fault tolerance,
3	GPU architecture and programming
4	Heterogeneity, interconnection topologies,
5	Load balancing,
6	Memory consistency model, memory hierarchies,
7	Message passing interface (MPI), MIMD/SIMD,
8	Multithreaded programming,
9	Parallel algorithms & architectures,
10	Parallel I/O, performance analysis and tuning,
11	Programming models (data parallel, task parallel, process-centric, shared/distributed memory),
12	Scalability and performance studies,
13	Scheduling, storage systems,
14	Synchronization,
15	Introduction to tools (Cuda, Swift, Globus, Condor, Amazon AWS, OpenStack, Cilk, gdb, threads, MPICH, OpenMP, Hadoop, FUSE)
16	Introduction to tools (Cuda, Swift, Globus, Condor, Amazon AWS, OpenStack, Cilk, gdb, threads, MPICH, OpenMP, Hadoop, FUSE)

CS435 - Software Quality and Testing

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Software Engineering

Description

The course aims to focus on exploring the essential concepts, methodologies, and practices related to ensuring the quality of software through effective testing.

Textbook(s)

- Paul C. Jorgensen, Byron DeVries. Software Testing A Craftsman's Approach, Fifth Edition, Auerbach Publications (2022), ISBN: 978-0367767624
- Andrew Pollner, Mark Fewster, Ina Schieferdecker,Test Automation Engineer Guide to the ISTQB Advanced Level Certification Rocky Nook,(2022), ISBN: 978-1937538088

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	To Understand how Software testing as a primary means to ensure software quality.	C2 1, 10
CLO 2	To discover other alternatives for quality assurance (QA), including defect prevention, process improvement, inspection, formal verification, fault tolerance, safety assurance, and damage control	C3 2, 6
CLO 3	Getting an understanding of the principles, procedure and protocol pertaining to the implementation of Agile Software Development methodology including its potential and impact	C3 3, 4, 8, 10
CLO 4	Understanding the measurement and analysis to close the feedback loop for quality assessment and quantifiable improvement.	C2 5, 6, 8

Week wise Lecture Schedule

W #	Lecture Topics
1	Basics of Software Testing – Important Concepts, Software Testing - Coverage, Limitations, Software Quality Engineering - Defect Prevention, Reduction, Containment.
2	Testing Object-oriented Software – What's Special? Testing Object-oriented Software, Models of the system, Test Model, Finite State Machines, Testing using FSMs.
3	Techniques for Test case generations using FSMs, The FREE State Model, Test generation using State Transition Trees
4	Testing using Class information, Testing using Combinational Models, Integration Testing for Model Based Software Testing , Regression Testing using Models
5	Model-based Test Data Generation, Test Oracles for MBT, Test Automation & Designing Test Harness for MBT, Test Adequacy Assessment
6	Mission of test groups, Generic Test Process, Test Concepts, What to Test: Black Box vs. White Box, Stopping Criteria: Coverage Based vs. Usage Based
7	An Initial Definition of (Software) Quality, Standard Definitions of (Software) Quality, Quality Problems in Large Software, Turning Failure to Success: SQA for Meeting Quality Expectations, The SQA Process
8	Different views of quality, Quality Frameworks, Defect Management,
9	Detailed Look at Major Testing Activities, Test Management: Responsibilities and Roles, Test Documentation
10	Fault Tolerance via recovery block (RB) and N version programming (NVP) , Safety Assurance Techniques/Strategies
11	FSM-Based Testing, Markov Chains as Enhanced FSMs, Unified Markov Models (UMM) for Testing
12	Quality Models and Measurements, Quality Models in QA Activities, Direct & Indirect Quality Measurements, Data Requirement by Models
13	Checklist-Based Testing, Partitions and Partition Testing, Usage-Based Testing OP Development
14	Ad hoc (non-systematic) testing, Checklist-based testing, Partition-based coverage testing, Boundary testing (BT), Control flow testing (CFT)
15	Quality Standards and Frameworks, Defect Handling in Quality Engineering. Software quality standards, ISO/IEC 25010, ISO/IEC 9126, Capability Maturity Model Integration (CMMI), IEEE 730, IEEE 1061, Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408)
16	Case studies

Specialization – Artificial Intelligence

CS431 - Programming for AI

Credit Hours: 3(2+1) **Prerequisites:** CC102
Course Category: Sp. Artificial Intelligence

Description

This course aims to introduce standard programming practices and to help develop programming skills necessary for designing and implementing Artificial Intelligence systems. The course introduces classical as well as modern state of the art programming language for Artificial Intelligence (Lisp, Prolog, Python, and R), and builds up the necessary programming background for the main courses like Machine Learning, Artificial Neural Networks & Deep Learning, Natural Language Processing, and Speech Processing. This course will help the students of Artificial Intelligence develop the programming acumen and style. The ultimate aim of this course is to help students in using the AI programming languages to solve problems of interest to them.

Textbook(s)

- Miller, B.N., Ranum, D.L. and Anderson, J., 2019. "Python programming in context." Jones & Bartlett Pub.

Other supplemental materials

- Joshi, P., 2017. "Artificial intelligence with python." Packt Publishing Ltd.
- Russell, S. and Norvig, P. "Artificial Intelligence. A Modern Approach", 3rd ed, Prentice Hall, Inc., 2015.

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand the fundamental constructs of Lisp, Prolog, and Python programming languages.	C2
CLO 2	Comprehend the fundamental constructs of programming languages for data analysis and representation.	C2
CLO 3	Understand and apply the Object-oriented concepts in the programming languages.	C2
CLO 4	Apply various libraries for plotting, interpreting and analyzing data in Python.	C3

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to Programming for AI, AI applications, Python setup with Anaconda, Jupyter Notebooks, Git basics
2	Python essentials recap, Control structures, Functions and modules, Clean code practices, AI-focused examples
3	Data structures for AI, Lists and dictionaries, List comprehensions, Nested structures, Dataset parsing
4	NumPy for array manipulation, Indexing and slicing, Broadcasting, Performance tuning, Image-as-array examples
5	Data visualization with Matplotlib, Seaborn basics, Histograms and scatter plots, Heatmaps, Feature visualization
6	Data handling with Pandas, Reading/writing files, DataFrame operations, Cleaning and transforming data
7	Object-Oriented Programming for AI, Classes and objects, Inheritance and polymorphism, Modeling ML components
8	Working with files and JSON, Reading/writing formats, Accessing APIs, Parsing AI-related data
9	Linear algebra for AI, Vectors and matrices, Dot product and norms, NumPy-based matrix operations
10	Intro to Scikit-learn, ML workflows, Data preprocessing, Pipelines, Hands-on with simple ML tasks
11	PyTorch/TensorFlow basics, Tensors and gradients, Building basic networks, Linear regression implementation
12	Image processing with OpenCV and PIL, Reading and transforming images, Preprocessing for vision tasks
13	Data pipelines for AI, Custom dataset creation, Batch loading, Image augmentation techniques
14	Handling large datasets, Memory-efficient loading, Profiling code, Optimizing data pipelines
15	Case Study/Project/Demo
16	Presentation/Peer Review/Q&A Sessions/Demo

CS343 - Knowledge Representation & Reasoning

Credit Hours: 3(2+1) **Prerequisites:** None
Course Category: Sp. Artificial Intelligence

Description

Knowledge representation is one of the fundamental areas of Artificial Intelligence. It is the study of how knowledge about the world can be represented and manipulated in an automated way to enable agents to make intelligent decisions. This course will provide an overview of existing knowledge representation frameworks developed within AI including but not limited to propositional and first-order logic, ontologies, planning, reasoning and decision making under uncertainty. The assignments component of the course would provide hands-on experience of software like Prolog, Protégé, probabilistic reasoning APIs and tools to support complex decision making. It is expected that after completing this course, students will understand (a) the foundations of Knowledge Representation & Reasoning and (b) which tools and techniques are appropriate for which tasks.

Textbook(s)

- Friedrich Bobach, Thomas Meyer, and Thomas Schaub, Knowledge Representation and Reasoning, Springer, 2020.

Other supplemental materials

- Ronald Brachman and Hector Levesque, Knowledge Representation and Reasoning, Morgan Kaufmann, 2004.

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand the fundamentals of knowledge representation and reasoning in deterministic situations	C2
CLO 2	Understand the challenges in representing knowledge and reasoning under uncertainty	C2
CLO 3	Analyze different situations and apply appropriate knowledge representation frameworks	C4
CLO 4	Development of hybrid approaches by synergizing the existing framework to solve complex decision-making problem	C4

Week wise Lecture Schedule

W #	Lecture Topics
1	Propositional Logic, Syntax and Semantics, Truth Tables
2	First-order Logic, Quantifiers, Logical Equivalences
3	Horn Clauses, Resolution in Propositional and FOL
4	Description Logic, Concepts and Roles, Basic Syntax
5	Reasoning using Description Logic, Concept Satisfiability
6	Forward Chaining, Backward Chaining, Rule-based Inference
7	Inference Engines, RETE Algorithm, Conflict Resolution
8	Semantic Networks, Concept Hierarchies, Frame-Based Systems
9	Ontologies, OWL, Ontology Languages, RDF
10	Logical Agents, Knowledge-Based Agents, Perception and Action
11	Planning, STRIPS, Partial Order Planning
12	Rule-Based Knowledge Representation, Production Rules
13	Reasoning Under Uncertainty, Probabilistic Logic
14	Bayesian Networks, Representation, Conditional Independence
15	Case Study/Project/Demo
16	Presentation/Peer Review/Q&A Sessions/Demo

CS441 - Computer Vision

Credit Hours: 3(2+1) **Prerequisites:** None
Course Category: Sp. Artificial Intelligence

Description
With a single glance a human interprets the entire scene. How many objects are present in the scene and where they are located. Which person is present in the scene. What will happen next. However, computers lack this capability. We have seen only face detectors so far working in our mobile phones. What is the challenge in understanding the 3D scene, i.e., the identity, the location and the size of the objects present in the scene. In this course we will introduce the basic concepts related to 3D scene modelling from single view and multiple views.

Textbook(s)
Computer Vision: Algorithms and Applications, by Richard Szeliski.

Other supplemental materials
- Multiple View Geometry in Computer Vision, by Richard Hartley and Andrew Zisserman.
- Computer Vision: A Modern Approach, by David Forsyth and Jean Ponce.
- Digital Image Processing, by Rafael Gonzalez and Richard Woods.

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understanding the single view geometry concepts	C2
CLO 2	Understanding the multiple view geometry concepts	C2
CLO 3	Apply concepts of CV for solving real world problems	C3

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to Computer Vision, Problems in CV, History and Modern Advancements
2	Fundamentals of Image Processing, Pixel operations, Color models
3	Image Filtering, Convolution, Gaussian and Median Filters
4	Image Pyramids, Fourier Transform in Image Processing
5	Hough Transform, Line and Circle Detection
6	Camera Models, Intrinsic and Extrinsic Parameters
7	Camera Looking at a Plane, Horizon Line, Rotation about Camera Center
8	Transformation Estimation, Point Correspondences, Homographies
9	2D and 3D Transformations, Rotation Matrices, Euler Angles
10	Edge Detection, Canny, Sobel Operators
11	Corner Detection, Harris Corner Detector, Shi-Tomasi Method
12	Feature Descriptors, HoG, SIFT, SURF
13	Deep Learning for CV: Object Detection (RCNN, YOLO, RetinaNet)
14	Deep Learning for CV: Segmentation (UNet, MaskRCNN), Image Generation (GANs)
15	Case Study/Project/Demo
16	Presentation/Peer Review/Q&A Sessions/Demo

CS442 - Natural Language Processing

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Artificial Intelligence

Description
Natural Language Processing (NLP) is the application of computational techniques to the analysis and synthesis of natural language and speech. This course is an introduction to NLP with prior programming experience in Python.

Textbook(s)
- Daniel Jurafsky and James H. Martin. 2018. Speech and Language Processing: An Introduction to Natural Language Processing. Third Edition. Prentice Hall

Other supplemental materials
Foundations of Statistical Natural Language Processing, Manning and Schütze, MIT Press. Cambridge, MA: May 1999

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand techniques for information retrieval, language translation, and text classification.	C2
CLO 2	Understand the advantages of using standard corpora. Identify examples of current corpora for a variety of NLP tasks.	C2
CLO 3	Understand and contrast deterministic and stochastic grammars, providing examples to show the adequacy of each	C2
CLO 4	Solve classic and stochastic algorithms for parsing natural language.	C3

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction & History of NLP, NLP applications, Challenges in NLP
2	Parsing algorithms, Constituency and dependency parsing
3	Basic Text Processing, Tokenization, Normalization, Lemmatization, Stopword removal
4	Minimum Edit Distance, Levenshtein Distance, Applications in Spell Checking
5	Language Modeling, N-grams, Smoothing, Backoff techniques
6	Spelling Correction, Noisy Channel Model, Edit probabilities
7	Text Classification, Naive Bayes, Logistic Regression for text
8	Deterministic and Stochastic Grammars, Context-Free Grammars (CFGs)
9	Representing Meaning, Semantic roles, Word Sense Disambiguation
10	Semantics and Vector Models, Word2Vec, GloVe, Embeddings
11	Sentiment Analysis, Polarity detection, Lexicon-based and ML approaches
12	Temporal Representations, Tense, Aspect, Event ordering
13	Corpus-based Methods, POS Tagging, Morphology
14	Information Retrieval, Vector Space Model, Precision and Recall
15	Case Study/Project/Demo
16	Presentation/Peer Review/Q&A Sessions/Demo

CS443 - Deep Learning

Credit Hours: 3(2+1) **Prerequisites:** None
Course Category: Sp. Artificial Intelligence

Description

This course will introduce Artificial Neural Networks and Deep Learning. ANN's basic architecture and how they mimic the human brain using simple mathematical models. Many of the important concepts and techniques around brain computing and the major types of ANN will also be introduced. Emphasis is made on the mathematical models, understanding learning laws, selecting activation functions and how to train the networks to solve classification problems. Deep neural networks have achieved state of the art performance on several computer vision and speech recognition benchmarks. This course will further build on the fundamentals of Neural networks and artificial intelligence and will introduce advanced topics in neural networks, convolutional and recurrent network structures, deep unsupervised and reinforcement learning.

Textbook(s)

Neural Network Design, 2nd Edition, Martin T. Hagan, Howard, B. Demuth, Mark Hudson Beale and Orlando De Jesus, Publisher: Martin Hagan; 2 edition (September 1, 2014), ISBN-10: 0971732116

Other supplemental materials

An Introduction to Neural Networks, James A Anderson, Publisher: A Bradford Book (March 16, 1995), ISBN-10: 0262011441

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand the fundamentals of neural networks in AI	C2
CLO 2	Explain how simple ANNs can be designed	C2
CLO 3	Apply ANN for classification Problems	C3
CLO 3	Apply deep learning algorithms to real-world problems	C3
CLO 4	Analyze results from deep learning to select appropriate solutions	C4

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction and history of neural networks, Applications in AI
2	Basic architecture of neural networks, Perceptron, Adaline, Minimum Error Learning
3	Machine learning theory, Training vs. test sets, Evaluation metrics
4	Gradient descent (Delta) rule, Hebbian learning, Neo-Hebbian learning
5	Differential Hebbian Learning, Drive Reinforcement Theory
6	Kohonen Self-Organizing Maps, Associative memory, Bi-directional associative memory (BAM)
7	Energy surfaces, Boltzmann machines, Theory of generalization
8	Backpropagation algorithm, Multi-layer perceptrons
9	Feedforward networks, Deep convolutional networks
10	Computational complexity of neural networks, Optimization techniques
11	Autoencoders, Sparse coding, Unsupervised deep learning
12	Deep belief networks, Restricted Boltzmann Machines
13	Deep recurrent neural networks, BPTT, LSTM, GRU
14	GPU programming for deep learning, CuDNN basics
15	Case Study/Project/Demo
16	Presentation/Peer Review/Q&A Sessions/Demo

CS444 – Evolutionary Computing & Swarm Intelligence

Credit Hours: 3(2+1) **Prerequisites:** None
Course Category: Sp. Artificial Intelligence

Description
Swarm Intelligence is a field of Artificial Intelligence inspired by the collective behavior of decentralized, self-organizing systems such as ant colonies, bird flocks, and fish schools. This course explores both natural and artificial swarm systems, focusing on the emergent behavior that arises from simple rules followed by individual agents. Students will study classical algorithms like Ant Colony Optimization (ACO) and gain hands-on experience in designing and analyzing swarm-based systems, including applications in robotics and optimization.

Textbook(s)
- Dorigo, M., & Stuetzle, T. Ant Colony Optimization, MIT Press, 2004. ISBN: 9780262042192
- Yang, X.-S. Nature-Inspired Optimization Algorithms, Elsevier, 2020. ISBN: 9780128196674

Other supplemental materials
- Bonabeau, E., Dorigo, M., & Theraulaz, G. Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, 1999. ISBN: 9780195131598

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Explain the foundational principles of swarm intelligence and its natural inspirations.	*C2*
CLO 2	Apply swarm-based algorithms such as ACO to solve optimization problems.	C3
CLO 3	Design and simulate swarm robotic systems with features like task allocation and fault tolerance.	C4
CLO 4	Analyze and evaluate the performance of swarm intelligence algorithms in various domains.	C5

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to Swarm Intelligence and key concepts (decentralization, self-organization)
2	Biological inspiration: ants, birds, fish, termites, and social insects
3	Swarm-based problem solving in natural systems
4	Travelling Salesman Problem (TSP) and its significance in optimization
5	Introduction to Ant Colony Optimization (ACO)
6	Construction graph, pheromone trails, and probability transition rules in ACO
7	Variants and enhancements of ACO
8	Applications of ACO in routing, scheduling, and other combinatorial problems
9	Introduction to swarm robotics and its distinction from classical robotics
10	Robot-to-robot communication and local decision-making
11	Division of labour and task allocation strategies in swarm robotics
12	Fault detection and resilience in swarm systems
13	Case studies in swarm robotics and collective transport
14	Simulation tools and platforms for swarm robotics (e.g., Webots, ARGoS)
15	Case Study/Project/Demo
16	Presentation/Peer Review/Q&A Sessions/Demo

CS445 - Reinforcement Learning

Credit Hours: 3(3+0) **Prerequisites:** AI322
Course Category: Sp. Artificial Intelligence

Description

Reinforcement Learning (RL) is a foundational area in Artificial Intelligence that enables autonomous agents to learn optimal behaviors through interaction with their environment. This course provides a comprehensive introduction to RL concepts and methods, including Markov decision processes, value-based and policy-based learning, model-free algorithms like Q-learning, function approximation, and exploration strategies. Students will also explore cutting-edge developments in deep reinforcement learning and apply their knowledge through coding assignments and a final project.

Textbook(s)

Sutton, R. S., & Barto, A. G. Reinforcement Learning: An Introduction, 2nd Edition. Available free: http://incompleteideas.net/book/the-book-2nd.html

Other supplemental materials

Handouts and research articles may also be used by the instructor.

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Explain the foundational principles of reinforcement learning, including Markov decision processes.	C4
CLO 2	Apply model-free RL methods like Q-learning and SARSA to solve decision-making tasks.	C3
CLO 3	Analyze policy iteration and value iteration techniques for policy optimization.	C6
CLO 4	Implement RL algorithms using function approximation and explore advanced methods like imitation learning.	C5

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to Reinforcement Learning; Real-world applications (Robotics, Healthcare, Games)
2	Agent-Environment Interface; Goals and Rewards
3	Markov Processes and Markov Decision Processes (MDPs)
4	Policies, Returns, and Value Functions
5	Dynamic Programming: Policy Evaluation and Policy Iteration
6	Value Iteration; Bellman Optimality Equation
7	Monte Carlo Methods for Prediction and Control
8	Temporal Difference Learning: TD(0), TD(λ)
9	Model-Free Control: Q-Learning and SARSA
10	Exploration vs. Exploitation; ε-greedy, Softmax Strategies
11	Function Approximation for RL: Linear Models, Tile Coding, and Gradient Descent
12	Policy Gradient Methods and Actor-Critic Algorithms
13	Imitation Learning and Learning from Demonstrations
14	Batch Reinforcement Learning and Off-Policy Learning, Introduction to Deep Reinforcement Learning; Deep Q-Networks (DQN)
15	Case Study/Project/Demo
16	Presentation/Peer Review/Q&A Sessions/Demo

Specialization – Cyber Security

CS351 - Wireless and Mobile Security

Credit Hours: 3(3+0) **Prerequisites:** CC261
Course Category: Sp. Cybersecurity

Description

The main objective of this course is to let the students learn the art of Cybersecurity and Network Protection. Throughout the journey, we will delve into various topics, including Wireless Networks, Mobile Device Security, Application Level Security, Ubiquitous Computing, IoT Security, Data Center Operations, and Cloud Computing. By mastering encryption techniques, understanding security challenges, and implementing robust measures, students will become skilled defenders in the ever-evolving cyber landscape. Join us now to strengthen your cybersecurity prowess and make a significant impact in the world of digital security.

Textbook(s)

Pallapa Venkataram, Satish Babu. Wireless and Mobile Network Security, 1st Edition, McGraw Hill (2010), ISBN: 978-0070700246
Jim Doherty. Wireless and Mobile Device Security, Jones and Bartlett Publishers, Inc (2021) ISBN: 978-1284211726

Other supplemental materials

Frank Adelstein, K.S.Gupta. Fundamentals of Mobile and Pervasive Computing. 1st Edition, Tata McGraw Hill (2005), ISBN: 978-0070603646

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand the fundamentals of Wireless Networks and Mobile Device Security, including wireless communication components, architectures, and mobile device security features.	C2 1,2,10
CLO 2	Identify and evaluate security risks associated with Wireless Networks, Mobile Apps, Ubiquitous Computing, and IoT, along with the implementation of security countermeasures.	C4 2,3,4
CLO 3	Comprehend various encryption protocols for securing Wireless Networks, Mobile Data, Cloud Computing, and IoT devices, as well as key management for ensuring data confidentiality and integrity.	C2 1,4,10
CLO 4	Analyze the security challenges and solutions in Mobile Device Management (MDM), Bring Your Own Device (BYOD) policies, Application Level Security in Mobile Ad Hoc Networks (MANETs), Data Center Operations, and Cloud Computing to develop effective security recommendations	C4 3,4, 10

Week wise Lecture Schedule

W #	Lecture Topics
1	Introduction to Wireless Networks: Basics of Wireless Communication, Wireless Network Architectures and Components, Wireless Standards (Wi-Fi, Bluetooth, etc.)
2	Wireless Network Security Risks: Common Security Threats in Wireless Networks, Wireless Eavesdropping and Data Interception
3	Wireless Encryption Protocols: WEP, WPA, WPA2, and WPA3 Security Protocols, Wi-Fi Protected Setup (WPS), Vulnerabilities
4	Mobile Device Security Fundamentals: Overview of Mobile Device Architecture, Mobile Operating System Security (Android, iOS)
5	Mobile App Security: Secure Mobile App Development Best Practices, Code Signing and App Distribution Security Mobile Device Management (MDM) and BYOD: Implementing MDM Solutions for Enhanced Security, Security Challenges of Bring Your Own Device (BYOD) Policies
6	Mobile Data Protection: Data Encryption on Mobile Devices, Remote Wiping and Tracking Lost Devices
7	Mobile Authentication and Biometrics, Multi-factor Authentication for Mobile Devices, Biometric Security on Smartphones
8	Wireless Intrusion Detection and Prevention: Deploying Wireless IDS/IPS Solutions, Detecting Rogue Access Points and Attacks
9	Application Level Security in MANETs: MANETs, Some applications of MANETs, MANET Features, Security Challenges in MANETs,
10	Application Level Security in MANETs: Security Attacks on MANETs, External Threats for MANET applications, Internal threats for MANET Applications, Some of the Security Solutions.
11	Ubiquitous Computing, Need for Novel Security Schemes for UC, Security Challenges for UC, and Security Attacks on UC networks, Some of the security solutions for UC.
12	IoT Security challenges, Standards and countermeasures
13	Data Center Operations - Security challenge, implement. Five Principal Characteristics of Cloud Computing
14	Data center Security Recommendations. Encryption for Confidentiality and Integrity, Encrypting data at rest, Key Management Lifecycle
15	Cloud Encryption Standards
16	Case Studies and Recent Trends

CS352 - Secure Software Design and Development

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Cybersecurity

Description

The module aims to develop core competencies in the fields of Secure Software Concepts, Secure Software Requirements, Secure Software Design, Secure Software Implementation/Coding, and Secure Software Testing. The course details the software security activities that need to be incorporated throughout the software development lifecycle. It provides comprehensive coverage that includes the people, processes, and technology components of software, networks, and host defenses.

Textbook(s)

- Gary McGraw. Software Security: Building Security In, 1st Edition, (2006), Addison-Wesley Professional ISBN: 978-0321356703
- Loren Kohnfelder. Designing Secure Software: A Guide for Developers, 978-1718501928, (2021), ISBN: 978-1718501928

Other supplemental materials

- Official (ISC)2 Guide to the CSSLP (latest)

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	A good comprehension of software security standards, models, processes and best practices that need to be incorporated throughout the software development lifecycle.	C2 1, 2, 10
CLO 2	Identify insecure programming patterns and the ability to replace them with secure alternatives.	C3 2, 3
CLO 3	Know tools for software security analysis and testing, and the ability to use them in practice and understand their capabilities and limitations.	C3 4, 5, 6, 10

Week wise Lecture Schedule

W #	Lecture Topics
1	Secure software concepts: System issues, System properties, Software Project Time Management, Software Project Costing, Software Quality Assurance
2	Security Concepts in the SDLC: Risk management, Security standards (e.g., coding standards, NIST standards, Federal Information Processing Standards);
3	Best practices: (e.g., OWASP development guide, OWASP code review guide, OWASP testing guide);
4	Security methodologies: (e.g., Socratic Methodology, Operationally Critical Threat, Asset, and Vulnerability Evaluation, STRIDE and DREAD, Open Source Security Testing Methodology Manual);
5	Security frameworks: (e.g., Zachman Framework, Control Objectives for Information and Related Technology, Sherwood Applied Business Security Architecture (SABSA));
6	Regulations- Privacy and Compliance;
7	Security Models: (e.g., BLP Confidentiality Model, Clark and Wilson Model (Access Triple Model));
8	Trusted Computing; Secure Software Requirements (Sources for Security Requirements, Types of Security Requirements);
9	Secure Software Design: (Design consideration, Information Technology Security Principles and Secure Design, Designing Secure Design Principles), Design Processes
10	Secure Software Implementation/Coding: Software Development Methodologies, Common Software Vulnerabilities and Controls
11	Secure Software Implementation/Coding: Defensive Coding Practices—Concepts and Techniques
12	Code Vulnerabilities and Avoiding Polymorphic Malware Attacks: : Buffer overflow, Format string, bug, Code vulnerabilities SQL Injection
13	Code Vulnerabilities and Avoiding Polymorphic Malware Attacks: Cross-site Scripting, Cross-site Request Forgery, Session management, Replication of vulnerabilities and exploitation;
14	Secure Software Testing: Security Testing Methodologies, Software Security Testing, Software Acceptance
15	Secure Software Testing: Legal Protection Mechanisms, Software Deployment- Operations-Maintenance and Disposal.
16	Case Studies

CS353 - Digital Forensics and Incident Response

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Cybersecurity

Description
This course is an introduction to computer forensics and investigation. It provides an understanding of how to conduct investigations to correctly gather, analyze and present digital evidence to different audiences. It also outlines the tools to locate and analyze digital evidence on a variety of devices, how to keep up to date with changing technologies, and laws and regulations in digital forensics.

Textbook(s)
- John Sammons. The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics, 2nd Edition, Elsevier, (2014)ISBN: 9780128016350
- Gerard Johansen. Digital Forensics and Incident Response : Incident Response Techniques and Procedures to Respond to Modern Cyber Threats, 2nd Edition, Packt Publishing (2020)ISBN: 978-1838649005

Other supplemental materials
- Joakim Kävrestad. Guide to Digital Forensics : A Concise and Practical Introduction, Springer (2017), ISBN: 978-3319674490

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	To develop knowledge about forensic law, standards, regulations and ethical values	C2 1, 8, 9 ,10
CLO 2	To be able to conduct digital forensics for multiple platforms and applications by various tools	C3 2, 5, 6
CLO 3	To be able to generate reports based on digital forensic tools for security systems and platforms	C3 3, 4, 6

Week wise Lecture Schedule

W #	Lecture Topics
1	An introduction to Digital Forensics; use of digital forensics; Key technical concepts;
2	Challenges in Digital Forensics ; The Difference between Computer Experts and Digital Forensics Experts;
3	Investigative Process Methodologies ; Education, Training, and Awareness;
4	Laws, Standards, and Regulations; Ethics and Professional Conduct;
5	Digital Evidence Management; Collecting evidence; Antiforensics;
6	Network forensics;
7	Mobile and Embedded Forensics;
8	Cloud forensics;
9	Internet Forensics;
10	Social media forensics;
11	Investigation Methods for Collecting Digital Evidence; Digital Forensic Readiness;
12	Digital forensics tools; Discovery of Computers and Storage Media;
13	Discovery of Audio/ Video Evidence;
14	Data Visualization; Data Sources; Graphing and Charting;
15	Analyzing Data; Data Distributions; Analysis Scenarios; Data Visualization Tools.
16	Case Studies

CS452 – Applied Cryptography [todo]

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Cybersecurity
By the end of this course the student will be able to:
• Understand the basics of cryptography, cryptanalysis, number theory, algebra, and complexity theory. (BT2 – Understand) (PLO1)
• Understand cryptographic protocols and algorithms in different application areas (BT2 – Understand)(PLO1)
• Evaluate security and performance of cryptographic algorithms and propose improvements (BT5 – Evaluate)(PLO4)
• Implement fundamental public key and symmetric key cryptographic algorithms. (BT3 – Apply) (PLO5)

CS453 – Usable Security and Policy [todo]

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Cybersecurity
By the end of this course the student will be able to:
• Explain foundational principles of usable security and privacy, including laws, PETs, and HCI methods. (BT2- Understand) (PLO1)
• Analyze usability challenges and evaluate research, tools, and design strategies in privacy and security mechanisms. (BT5- Evaluate) (PLO3)
• Apply user-centered design and HCI methods to assess and improve usability of privacy and security solutions. (BT3- Apply) (PLO5)
• Develop and present a prototype that integrates usability principles with privacy and security by design. (BT6 – Create) (PLO4)

CS454 – Cloud Security [todo]

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Cybersecurity

By the end of this course the student will be able to:
• Describe security challenges in cloud environments. (BT2 – Understand) (PLO1)
• Apply identity, access, and encryption controls in cloud systems. (BT3 – Apply)(PLO5)
• Evaluate cloud service providers' compliance with security standards. (BT5 – Evaluate)(PLO4)

CS455 – AI for Cyber Security [todo]

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Sp. Cybersecurity
By the end of this course the student will be able to:
• Explain core cybersecurity concepts and the role of ML/DL in addressing modern threats. (BT2-Understand) (PLO1)
• Analyze attack vectors and assess ML/DL techniques for detection and prevention. (BT4 – Analyze) (PLO3)
• Critique traditional vs. ML/DL approaches in terms of accuracy, adaptability, and ethics. (BT5 – Evaluate) (PLO4)

Fundamental Computing or Mathematical Courses

CS101 – Application of ICT [todo]

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Mandatory (G. Edu.)

CS102 – Applied Mathematics for Computing I

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Mandatory (QRI)

Description

This course is designed to provide a strong mathematical foundation for computing students with diverse academic backgrounds. Students entering with HSSC without Mathematics are offered this compulsory course to build essential quantitative reasoning skills. Likewise, students who have studied Mathematics at the HSSC level are often unfamiliar with how mathematical concepts are applied in computing disciplines.

The course is therefore mandatory for both categories to ensure reinforcement of fundamental mathematical concepts and to develop problem-solving skills relevant to computer science. The content is aligned with the Quantitative Reasoning requirements of the curriculum and emphasizes conceptual understanding and practical application in computing contexts.

Textbook(s)

- Mathematics for Computer Science by Eric Lehman, F. Thomson Leighton, and Albert R. Meyer (MIT Press)

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand and explain fundamental mathematical concepts including real numbers, complex numbers, matrices, sequences, series, and basic combinatorics relevant to computing.	C2 1, 2
CLO 2	Apply mathematical methods such as matrix operations, system of linear equations, differentiation, and integration to solve computational and logical problems.	C3 2, 3
CLO 3	Analyze and solve mathematical problems arising in computing contexts and interpret the results using appropriate mathematical reasoning and techniques.	C3 4, 10

Week wise Lecture Schedule

W #	Lecture Topics
1	Revision of algebraic properties of real numbers and order properties of real numbers.
2	Imaginary unit , imaginary numbers, and integer powers
3	Complex numbers, operations on complex numbers, and algebraic properties of complex numbers.
4	Conjugate of a complex number and its properties, absolute value (modulus) of a complex number and its properties.
5	Determination of real and imaginary parts of complex numbers and polar form of complex numbers.
6	Matrices, types of matrices, operations of matrices, and algebraic properties of matrices.
7	Determinant of a matrix and its properties, adjoint and multiplicative inverse of a matrix.
8	Echelon form and reduced echelon form, finding multiplicative inverse of a matrix using row operations.
9	Rank of a matrix, consistency of systems of linear equations, applications of echelon and reduced echelon forms, and review of matrices and determinants.
10	Solution of systems of equations in three unknowns by matrix inversion method and Cramer's rule.
11	Gaussian elimination method, introduction to sequences and series, arithmetic sequence and arithmetic series.
12	Geometric sequence and geometric series (finite and infinite), harmonic sequence.
13	Factorial of a positive integer, fundamental principle of counting, permutations of distinct objects and objects not all different.
14	Combinations and mathematical induction.
15	Binomial theorem, binomial series, trigonometric ratios, and trigonometric functions.
16	Advanced applications of binomial theorem and trigonometric identities, comprehensive review of course contents, problem-solving sessions, and course conclusion.

CS103 – Applied Mathematics for Computing II

Credit Hours: 3(3+0) **Prerequisites:** CS101
Course Category: Mandatory (QRII)

Description

This course is designed to provide a strong mathematical foundation for computing students with diverse academic backgrounds. Students entering with HSSC without Mathematics are offered this compulsory course to build essential quantitative reasoning skills. Likewise, students who have studied Mathematics at the HSSC level are often unfamiliar with how mathematical concepts are applied in computing disciplines. It focuses on advanced mathematical concepts with direct applications in computing. Students learn how to model and analyze problems using functions, limits, derivatives, integrals, and series.

The course is therefore mandatory for both categories to ensure that students develop the ability to apply mathematical reasoning to computational problems, understand the connection between differentiation and integration, and use these concepts in algorithm design, data analysis, and problem-solving in computer science contexts. The content is aligned with the Quantitative Reasoning requirements of the curriculum and emphasizes conceptual understanding and practical application in computing contexts.

Textbook(s)

- Advanced Engineering Mathematics by Erwin Kreyszig

Course Learning Objectives

#	Course Learning Outcomes (CLOs)	BT/ PLO #
CLO 1	Understand and explain advanced concepts of functions, limits, continuity, differentiation, integration, and series, and their relevance to computing.	C2 1, 2
CLO 2	Apply differentiation and integration techniques, sequences and series, and other advanced mathematical methods to solve computational problems.	C3 2, 3
CLO 3	Analyze and interpret mathematical problems arising in computing and provide solutions using appropriate mathematical methods and reasoning.	C3 3, 10

Week wise Lecture Schedule

W #	Lecture Topics
1	Functions, one-one correspondence, and methods of expressing relations between variables x and y: explicit, implicit, and parametric forms.
2	Graphs of functions, even and odd functions, piece-wise defined functions, and absolute value function.
3	Algebraic functions and transcendental functions, combining functions including sum, difference, product, quotient, and composition of functions.
4	Limit of a function, left-hand and right-hand limits, and basic concepts of limits.
5	Limit laws, limits of algebraic and transcendental functions, and limits involving infinity.
6	Continuous functions, properties of continuous functions, and composition of continuous functions.
7	Derivative of a function, rules of differentiation, and derivatives of algebraic functions.
8	Chain rule, implicit differentiation, and derivatives of transcendental functions.
9	Applications of derivatives including tangent and normal to a curve, rate of change problems, and review of differentiation techniques.
10	Higher-order derivatives and applications of higher derivatives.
11	Taylor series and Maclaurin series of functions.
12	Differentials of variables, integration as an anti-derivative, and theorems of anti-derivatives.
13	Integration using substitution, rules of integration, and integrals of algebraic and transcendental functions.
14	Definite integrals, fundamental theorem of calculus, and properties of definite integrals.
15	Applications of definite integrals, including area calculation.
16	Applications of integration, conceptual linkage between differentiation and integration, comprehensive revision of course contents, and course conclusion.

Specialization – Professional Courses

CS496 – Professional Certification

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Mandatory Certificate

Description
A certification will be considered equivalent to 1 credit hour if it comprises at least 16 hours of study. The department has prepared a list of recommended certifications given in this course. The department will guide students in selecting relevant certifications, ensuring alignment with current market needs and the program's objectives.

S.No	Course Title	Platform/ Issuer
1.	*Microsoft Certified: Azure Developer Associate*	*Microsoft*
2.	*AWS Certified Developer – Associate*	*AWS*
3.	*Professional Scrum Master (PSM)Certification*	*Scrum*
4.	*Oracle Certified Professional, Java Application Developer Certificate*	*Oracle*
5.	*PCEP™ – Certified Entry-Level and Professional Python Programmer*	*PCEP*
6.	*Google Advanced Data Analytics Certificate*	*Google*
7.	*IBM Full Stack Software Developer Professional Certificate*	*Coursera*
8.	*Foundations of Human-Computer Interaction*	*Stanford Online*
9.	*Cognitive Psychology for Interface Design*	*Iowa State Uni*
10.	*User-Centered Design and Prototyping*	*Inter Design Foundation*
11.	*UI/UX Design Specialization*	*Coursera*
12.	*Usability Eng. and Evaluation*	*MIT*
13.	*HCI for UX Design*	*Coursera*
14.	*Digital Product Management*	*Coursera*
15.	*Game Design and Development*	*Coursera*
16.	*Cognitive Science and AI*	*Georgia Tech*
17.	*Design Thinking for Innovation*	*Coursera*
18.	*Intro to Probability and Data*	*Coursera*
19.	*Discrete Mathematics and Probability*	*edX*
20.	*Algorithms for Dummies Specialization*	*Coursera*
21.	*Python for Everybody Specialization*	*Coursera*
22.	*Introduction to Computer Science and Programming Using Python*	*edX*
23.	*Machine Learning Specialization*	*Coursera*
24.	*Deep Learning Specialization*	*Coursera*
25.	*Introduction to Deep Learning*	*edX*
26.	*Knowledge Graphs and Reasoning Systems*	*Udemy*
27.	*Optimization Methods in Applied Mathematics*	*Coursera*
28.	*Generative AI with Large Language Models*	*Coursera*
29.	*Practical Guide to Generative AI*	*edX*
30.	*Natural Language Processing Specialization*	*Coursera*
31.	*Natural Language Processing*	*edX*
32.	*Computer Vision Basics*	*Coursera*
33.	*Deep Learning Specialization*	*edX*
34.	*Reinforcement Learning Specialization*	*Coursera*
35.	*Deep Reinforcement Learning*	*edX*
36.	*Microsoft Azure AI Fundamentals*	*Microsoft*
37.	*AWS Cloud Quest: Cloud Practitioner*	*AWS*
38.	*Hugging Face: NLP Fundamentals*	*Hugging Face*
39.	*Tableau Desktop Specialist Certification*	*Tableau*
40.	*AWS Certified Machine Learning – Specialty*	*AWS*
41.	*Google Cloud Professional Machine Learning Engineer*	*Google*
42.	*Microsoft Certified: Azure AI Engineer Associate*	*Microsoft*
43.	*TensorFlow Developer Certificate*	*Google*
44.	*Certified Analytics Professional*	*INFORMS*
45.	*Oracle Certified Professional, Java SE 8 Programmer*	*Oracle*
46.	*Unity Certified Game Development by EC Council - UCA 156*	*Unity*
47.	*Microsoft Power BI Data Analyst Professional*	*Microsoft*
48.	*Google Project Management: Professional Certificate*	*Google*

CS497 – Internship

Credit Hours: 3(3+0) **Prerequisites:** None
Course Category: Mandatory

Description
An internship for at least nine (9) continuous weeks is acceptable.
Host institutions could be local, provincial, or national government offices; an autonomous body (attached department of a government agency); a business enterprise (industrial and commercial enterprises); an academic institution (university, college, or school; public or private); or a registered NGO.

End of the Document

This document is approved by the Board of Studies meeting in 2026 for the department of Computer Science at AWKUM.