Ministry of Higher education and Scientific Research

UNIVERSITY OF ANBAR

COLLEGE OF COMPUTER SCIENCES AND INFORMATION TECHNOLOGY

Department of Computer Networks Systems

Department of Computer Networks Systems

CATALOUGE

2023/2024

Prepared by: ……..

Vision

Vision Statement for the Computer Networks Systems Department in a Computer Science and information technology:

"To be a leading Networking Department, empowering students to become skilled professionals in the rapidly evolving field of computer networking, enabling them to drive innovation and shape the future of technology."

Explanation:

1. Leading Networking Department: The vision emphasizes the desire to be at the forefront of networking education, setting the standard for excellence in the field. It reflects a commitment to staying updated with the latest advancements and best practices.

2. Empowering Students: The vision recognizes the importance of empowering students by providing them with a comprehensive education that equips them with the necessary knowledge, skills, and hands-on experience to excel in the networking industry.

3. Skilled Professionals: The vision focuses on producing graduates who are highly skilled and capable of meeting the demands of the evolving networking landscape. It highlights the aim to develop well-rounded professionals who can adapt to new technologies, troubleshoot complex network issues, and contribute to the growth of the industry.

4. Rapidly Evolving Field: The vision acknowledges the dynamic nature of the networking field. It signifies the department's commitment to keeping pace with emerging trends, technologies, and industry standards, ensuring that students receive an education that is relevant and up to date.

5. Shaping the Future of Technology: The vision highlights the department's aspiration to play a significant role in shaping the future of technology through its contributions to the field of computer networking. It signifies the intention to produce graduates who can make meaningful contributions and lead advancements in networking technologies and practices.

Mission

The mission of the Networking Department in our Computer College is to provide comprehensive and cutting-edge education in the field of networking. We aim to equip our students with the knowledge, skills, and practical experience necessary to excel in the rapidly evolving networking industry.

Our department is committed to fostering a dynamic learning environment that promotes innovation, collaboration, and critical thinking. We strive to empower our students to become competent and resourceful professionals who can meet the challenges of networking in today's interconnected world.

Key Principles:

1. Quality Education: We are dedicated to delivering high-quality education that meets industry standards and prepares our students for successful careers in networking. Our curriculum is regularly updated to reflect the latest advancements and emerging technologies in the field.

2. Practical Experience: We emphasize hands-on learning and practical experience to ensure that our students develop the necessary skills to design, implement, and troubleshoot networks. Through lab exercises, projects, and industry partnerships, we provide opportunities for real-world application of theoretical concepts.

3. Professional Development: We foster a culture of continuous learning and professional development among our students and faculty. We encourage participation in workshops, seminars, industry conferences, and certification programs to enhance technical expertise and stay abreast of industry trends.

4. Ethical Practices: We emphasize the importance of ethical behavior and responsible use of technology in networking. Our students are trained to prioritize privacy, security, and ethical

considerations in all their networking activities. We promote integrity, professionalism, and adherence to ethical guidelines.

5. Community Engagement: We actively engage with the local and global networking community to foster networking excellence and contribute to its advancement. We organize events, seminars, and conferences to facilitate knowledge sharing and networking opportunities for our students and faculty.

Programme Educational Objectives (PEOs)

The goals for the Networking Department in a computer college may vary depending on the specific objectives and priorities of the institution. However, here are some common program goals that a Networking Department might strive to achieve:

1. Provide comprehensive networking education: The Networking Department should aim to deliver a comprehensive curriculum that covers fundamental and advanced concepts in computer networking. This includes topics such as network protocols, network security, routing and switching, wireless networks, network management, and emerging networking technologies.
2. Develop practical skills: The department should focus on equipping students with hands-on skills that are applicable to real-world networking scenarios. Practical training should involve configuring and managing network devices, troubleshooting network issues, designing network infrastructures, and implementing network security measures.
3. Keep pace with industry trends and advancements: The field of networking is rapidly evolving, with new technologies, protocols, and trends emerging regularly. The Networking Department should strive to stay up-to-date with these advancements and incorporate relevant and cutting-edge topics into the curriculum. This ensures that students are equipped with the knowledge and skills needed to adapt to the ever-changing networking landscape.
4. Promote teamwork and collaboration: Networking professionals often work in teams and collaborate with colleagues to design, implement, and manage networks. The program should emphasize the importance of teamwork and

provide opportunities for students to work collaboratively on networking projects and assignments. This helps develop their interpersonal and communication skills, as well as their ability to work effectively in a team-based environment.

5. Prepare students for industry certifications: Many networking professionals pursue industry certifications to validate their skills and enhance their career prospects. The Networking Department should align the curriculum with relevant industry certifications, such as Cisco Certified Network Associate (CCNA) or CompTIA Network+, and provide resources and guidance to help students prepare for these certifications..
6. Ensure high-quality teaching and learning: The department should prioritize the recruitment and professional development of skilled faculty members who possess both industry experience and teaching expertise. Regular assessments and feedback mechanisms should be implemented to ensure the quality of teaching and learning experiences. Additionally, the program should leverage modern educational technologies and resources to enhance the learning environment.
7. Support lifelong learning: Networking professionals need to continuously update their knowledge and skills to keep pace with advancements in the field. The department should encourage and support students' lifelong learning by offering opportunities for professional development, such as continuing education programs, workshops, and seminars. This helps students stay relevant in their careers and adapt to the evolving demands of the networking industry.

Total credit hours of courses according to the adopted levels

Courses		Levels				Total
		First	Second	Third	Fourth	
University	Compulsory		---	---	---	
	Elective		4	2	4	10
Total			4	2	4	10
College	Compulsory		18	8	20	46
	Elective		--	--	--	
Total			18	8	20	46
Department	Compulsory		33	26	28	87
	Elective		1	2	1	4
Total			34	28	29	91
Total			56	38	53	147

Graduation Requirements

Requirements	Credit
University Requirements	5
College Requirements	9
Department Requirements	26
Elective Courses	4
Total	**44**

University Requirements: 13 credit hours

Course code	Course Title	Credit hours	Weekly hours	Prerequisite
	English Language-III	2	2	
	English Language-II	2	2	
	The crimes of the defunct Ba'ath	2	2	
	English Language-IIII	2	2	
	Research Methodology	1	2	
Total		**9**	**10**	

College Requirements: …… credit hours

Course Code	Course Title	Credit hours	Weekly hours			Prerequisite
			Lec.	Tut.	Lab	
CCIT060	Data Structures هياكل البيانات	3	2		2	
CCIT061	Object Oriented Programming I البرمجة الكيانية1	4	3		2	
CCIT062	Numerical Analysis التحليل العددي	3	2		2	
CCIT063	Object Oriented Programming II البرمجة الكيانية2	4	3		2	
CCIT064	Visual Programming, I البرمجة المرئية1	3	2		2	
CCIT065	Visual Programming, 2 البرمجة المرئية2	3	2		2	
CCIT066	Artificial Intelligence I1 الذكاء الاصطناعي	3	2		2	
CCIT067	Artificial Intelligence 2الذكاء الاصطناعي2	3	2		2	
CCIT068	Project in CNS مشروع في نظم شبكات الحاسوب	6	--		12	
Total		**32**	**18**		**28**	

Department Requirements: ..... credit hours

Course Code	Course Title	Credit hours	Weekly hours			Prerequisite
			Lec.	Tut.	Lab	
CNDC212	Advanced Mathematics	3	3			
CNDC204	Digital Electronic	3	2		2	
CNDC206	Microprocessors	3	2		2	
CNDC210	Data Communication	2	2			
CNDC218	Information Theory & Coding	2	2			
CNDC209	Algorithms	3	2		2	
CNDC205	Computer Architecture	2	2			
CNDC207	Computer Networks	3	2		2	
CNDC213	Web Design	3	2		2	
CNDC309	Database Management Systems	3	2		2	
CNDC306	Wireless Networks	3	2		2	
CNDC303	Web Programming	3	2		2	
CNDC308	Digital Signal Processing I	2	2			
CNDC301	Software Engineering	2	2			
CNDC302	Multimedia	3	2		2	
CNDC322	Distributed Database	3	2		2	
CNDC307	Network Programming	3	2		2	
CNDC313	Digital Signal processing II	2	2			
CNDC408	Network Protocols & Services	3	2		2	
CNDC413	Information Security	2	2			
CNDC412	Web Application Development I	3	2		2	
CNDC401	Operating Systems	3	2		2	

CNDC407	Mobile Computing	3	2		2	
Total		**62**	**47**		**30**	

Elective Courses

Course Code	Course Title	Credit hours	Weekly hours			Prerequisite
			Lec.	Tut.	Lab	
CNDE301	Software Engineering	2	2			
CNDE302	Multimedia	3	2		2	
CNDE429	Artificial Intelligence 2	3	2		2	
CNDE205	Computer Architecture	2	2			
Total		**10**	**8**		**4**	

Total Credits = 103
Total In-Touch Hours = 133

SECOND LEVEL

Course code	Course Title	Credit Hours	Weekly hours			Prerequisite
			Lec.	Tut.	Lab.	
CCIT060	**Data Structure**	4	2		2	
CNDC212	Advanced Mathematics	2	2			
CNDC204	Digital Electronic	4	2		2	
CNDC206	**Microprocessors**	4	2		2	
CNDC210	**Data Communication**	2	2			
CCIT061	Object Oriented Programming I	5	3		2	
UOA018	The crimes of the defunct Ba'ath	2	2			
UOA011	English2	2	2			
CNDC209	Algorithms	4	2		2	
CNDC062	Numerical Analysis	4	2		2	
CNDE205	Computer Architecture	2	2			
CNDC203	Computer Networks	4	2		2	
CNDC213	Web Design	4	2		2	
CCIT062	Object Oriented Programming II	5	3		2	
CNDC218	Information Theory & Coding	2	2			
Total		**50**	**32**		**18**	

THIRD LEVEL

Course code	Course Title	Credit Hours	Weekly hours			Prerequisite
			Lec.	Tut.	Lab.	
CCIT064	**Visual Programming, I**	4	2		2	
CNDC309	**Database Management Systems**	4	2		2	
CNDC306	**Wireless Networks**	4	2		2	
CNDC303	**Web Programming**	4	2		2	
CNDC308	**Digital Signal Processing I**	2	2			
UOA313	**English3**	2	2			
CNDE301	**Software Engineering**	2	2			
CCIT065	**Visual Programming II**	4	2		2	
CNDE302	**Multimedia**	4	2		2	

CNDC322	Distributed Database	4	2		2	
CNDC307	Network Programming	4	2		2	
CNDC313	Digital Signal processing II	2	2			
Total		40	24		16	

FOURTH LEVEL

Course code	Course Title	Credit Hours	Weekly hours			Prerequisite
			Lec.	Tut.	Lab.	
CNDC408	Network Protocols & Services	4	2		2	
CNDC413	Information Security	2	2			
CCIT066	Artificial Intelligence I	4	2		2	
CNDC412	Web Application Development I	4	2		2	
CNDC401	Operating Systems	4	2		2	
UOA019	Research Methodology	2	2			
UOA422	English4	2	2			
CNDC406	Switching and Routing in the Network	4	2		2	
CNDC404	Networks Security	2	2			
CCIT067	Artificial Intelligence II	4	2		2	
CNDC405	Web Application Development II	4	2		2	
CNDC407	Mobile Computing	4	2		2	
CCIT068	Project in CNS	12			12	
Total		52	24		28	

Curriculum Map of the PEOs with courses according to learning outcomes

the level	Course code	Course Title	Cognitive goals			Skills objectives of the program			Emotional and value goals		
			A1	A2	A3	B1	B2	B3	C1	C2	C3
Second Level	CCIT060	Data Structure	√				√		√		
	CNDC212	Advanced Mathematics	√				√		√		
	CNDC204	Digital Electronic	√				√		√		
	CNDC206	Microprocessors	√				√		√		
	CNDC210	Data Communication	√				√		√		
	CCIT061	Object Oriented Programming I	√			√			√		
	UOA018	The crimes of the defunct Ba'ath	√			√			√		
	UOA011	English2	√			√			√		
	CNDC209	Algorithms		√		√					√
	CCIT062	Numerical Analysis		√		√					√
	CNDC205	Computer Architecture		√				√			√
	CNDC063	Computer Networks		√				√			√
	CNDC213	Web Design			√		√				√
	CCIT063	Object Oriented Programming II			√		√				√
	CNDC218	Information Theory & Coding			√		√				√
The third level	CCIT064	Visual Programming, I		√		√				√	
	CNDC309	Database Management Systems		√			√			√	
	CNDC306	Wireless Networks		√		√				√	
	CNDC303	Web Programming		√		√					√
	CNDC308	Digital Signal Processing I		√			√			√	
	UOA313	English3	√			√		√	√		
	CNDC301	Software Engineering	√				√				√
	CCIT065	Visual Programming II	√			√		√			√

	CNDC302	Multimedia			√	√					√
	CNDC322	Distributed Database			√			√		√	
	CNDC307	Network Programming			√		√		√		
	CNDC313	Digital Signal processing II			√		√		√		
fourth level	**CNDC408**	Network Protocols & Services		√				√	√		
	CNDC413	Information Security		√		√			√		
	CCIT066	Artificial Intelligence I		√		√			√		
	CNDC412	Web Application Development I		√		√			√		
	CNDC401	Operating Systems			√	√				√	
	UOA019	Research Methodology	√			√				√	
	UOA422	English4	√				√				
	CNDC406	Switching and Routing in the Network						√			√
	CNDC404	Networks Security		√			√		√		
	CCIT067	Artificial Intelligence II	√				√			√	
	CNDC405	Web Application Development II	√					√		√	
	CNDC407	Mobile Computing			√		√		√		
	CCIT068	Project in CNS		√				√	√		
Elective courses	**CNDC205**	Computer Architecture		√		√				√	
	CNDE301	Software Engineering	√				√		√		
	CNDE302	Multimedia	√				√				√

UNIVERSITY REQUIREMENT COURSES

OR

College REQUIREMENT COURSES

OR

Department REQUIREMENT COURSES

Course Code: CNDC205
Course Title: Computer Architecture
Credit Hours: 2_2_0_0

- Course Definition

A comprehensive study of the structure and functions of computers, focusing on analyzing and understanding each part of computer components including the CPU, memory, and input/output units. The course also includes a review of the significant history of computer evolution, data representation in digital systems, as well as error detection and correction techniques. The curriculum also covers Boolean algebra and digital logic, along with an introduction to simple computer computing using the "MARIE" model.

- Course Topics:

- Introduction to computer components and historical review.
- Data representation in computer system.
- Error detection and correction.
- Boolean algebra and digital logic.
- Introduction to simple computer "MARIE".
- MARIE architecture.
- Instruction set architecture.
- Instruction types.
- Memory (1).
- Memory (2).
- Input/output storage system.
- System software.
- Alternative architecture.
- Embedded systems.
- Performance measurement and analysis.

- Course Description

The course covers computer components and their historical development, as well as data representation, Boolean algebra, digital logic, simple computer architecture, memory types, and many other topics.

- Learning Outcomes:

Students are expected to gain a deep understanding of computer components and their functions, skills in data representation and error handling, a profound understanding of Boolean algebra and digital logic,

and the ability to analyze and understand computer architecture and memory types.

- Recommended Textbooks:
"Computer Organization and Design" by David A. Patterson and John L. Hennessy, and "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson.

- Prerequisites:
Students are encouraged to have a good understanding of mathematical concepts and algebra, as well as basic knowledge of computer science, data structures, and programming.

- Lab. Topics: Not applicable

Course Code: [CNDC210]
Course Title: [Data Communications]
Credit Hours: [2_2_0_0]
- Course Definition
The course covers data communication processes across computer networks and telecommunications. It focuses on how data is exchanged between different devices over the internet and other communication networks, with an emphasis on the protocols and technologies used in these processes.

- Course Topics:
• Data Communications: Overview, TCP/IP Model
• Characteristics of Data Communications
• Data Representation
- Data Flow
- Data Representation
- Data and Signals
- Periodic and Aperiodic Signals
- Relationship between Frequency and Period
- Digital Signals
- Baud Rate
- Types of Channels
- Bandwidth
- Signal Bandwidth
- Channel Bandwidth
- Shannon Capacity
- Time and Frequency Domain Representation of Signals

- Conversion from Digital to Digital I: Linear Encoding, Linear Encoding Schemes
- Conversion from Digital to Digital II: Block Encoding, Modulation
- Digital Transmission: Conversion from Analog to Digital: Pulse Code Modulation (PCM), Pulse Code Modulation (DM)
- Transmission Modes: Parallel Transmission, Serial Transmission
- Analog Transmission: Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM)
- Transmission Media: Guided and Unguided
- Multiplexing: Frequency Division Multiplexing, Wavelength Division Multiplexing, Time Division Multiplexing.

- Course Description
The course covers the study of data transmission and communications across networks, including how to design, operate, and maintain networks efficiently and securely. The topics covered also include a deep understanding of various protocols and their practical applications.

- Learning Outcomes:
Students are expected to gain a deep understanding of the fundamentals of data transmission and network communications, skills in designing, operating, and managing networks efficiently and securely, as well as the ability to analyze and solve problems related to communications and networks.
- Recommended Textbooks:
"Data and Computer Communications" by William Stallings and "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross.

- Prerequisites :
It is preferable for students to have a good understanding of networking concepts and communication protocols, in addition to basic knowledge of programming techniques and operating systems.

- Lab. Topics:
None

Course Code: [CCIT060]
Course Title: [Example: Data Structures]
Credit Units : [3_2_0_2]

- Course Definition
The course aims to introduce students to the concepts and fundamentals of data structures and associated algorithms. It covers topics such as numbers, pointers, linear lists, linear list operations, and their applications, among others.

- Course Topics:
- Introduction to Data Structures
- Algorithms and Complexity
- Arrays and Pointers
- Linear Lists and Types of Linear Lists
- Stack Operations
- Stack Applications
- Queue Operations
- Queue Applications
- Review of Pointers and Structures
- Representation of Linked Lists
- Linked List Operations
- Basic Concepts of Trees
- Basic Concept of Graph
- Graph Traversal Algorithm
- Hashing

- Course Description
The course provides a comprehensive overview of data structures and algorithms. It focuses on the application of various data structures and their usage in solving complex programming problems.

- Learning Outcomes:
Students are expected to gain a deep understanding of the fundamentals of data structures and algorithms, as well as skills in designing and implementing data structures to solve complex programming problems.

- Recommended Textbooks:
"Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, and "Data Structures Using C" by Yedidyah Langsam, Moshe Augenstein, and Aaron M. Tenenbaum.

- Prerequisites :
Students are recommended to have prior knowledge of programming language and basic programming concepts, in addition to a basic understanding of algorithms.

- Lab. Topics:
Introduction to Data Structures: Arrays, Linked Lists, and Stacks
Implementation and basic operations on arrays, linked lists, and stacks in a programming language (e.g., C++, Java)
Analyzing the time and space complexity of operations on these data structures
Queues and Circular Queues: Implementation and basic operations
Priority Queues and Heaps: Implementation and basic operations
Trees: Binary Trees, Binary Search Trees, and AVL Trees
Implementation and basic operations on trees

Course Code: CNDC212
Advanced Mathematics (2-2-0-0)

- Designation as a „required" or „elective" course:
This is a required course for the computer science department .

- **Course Topics :**

- Abstract of differential equation
- Separable equation
- Solve some example
- Homogenous equation
- Exact equation
- Linear equation
- Some example
- Bernoulli equation
- Second order differential equation
- Some example
- Laplace transform
- Power series , Fourier series
- Mid exam
- Review
- Final exam

- Preparatory week before the final Exam

- **Course Description** :

1. Understand the concept of ordinary and partial.
2. Understand the method of solving the first order differential equation.
3. Understand the method of solving second order differential equation.
4. Understand the Laplace transform.
5. Understand the Fourier series.
6. Subject-specific skills:
7. Explain what mean of ordinary and partial.
8. Classify the method of solving.

- **Course Outcomes:**

- Understand the concept of ordinary and partial.
- Understand the method of solving the first order differential equation.
- Understand the method of solving second order differential equation.
- Understand the Laplace transform.
- Understand the Fourier series.
- Subject-specific skills:
- Explain what mean of ordinary and partial.
- Classify the method of solving.
- Classify the differential equation.
- Teaching and Learning Methods By solving many exercises.

- **Recommended Textbook(s):**
- **Prerequisites:** None
- **Lab. Topics :**

Course Code: CNDC204
Course Title: Digital Electronics
Credit Units : 3-2-0-2

- Course Definition

Digital Electronics course focuses on the study of designing and applications of digital circuits and systems. It includes understanding digital fundamentals, logic gates, digital circuits, and their applications in areas such as digital processing, communications, and control systems.

- Course Topics:

- Analog vs Digital
- Basics of Digital Electronics
- Logic Gates and Implementation of Combinational Circuits using Logic Gates
- Karnaugh Maps (Single Variable, Two Variables, Three Variables)
- Applications of Registers and Counters
- Arithmetic Circuits
- Binary Number System
- Counters (Asynchronous, Synchronous, Decimal, Up/Down, Ripple, Decoding Counters)
- Implementation of Sequential Circuits using Flip-Flops and Latches
- Data Converter Circuits (Analog to Digital and Digital to Analog Conversion)
- Microcontrollers: Introduction to Arduino
- Arduino Programming
- Designing Digital Circuits using Verilog

- Course Description:
This course provides a comprehensive overview of designing and analyzing digital circuits and digital electronics systems. The course focuses on the practical applications of digital electronics in various fields, particularly in computing, communications, and control.

- Learning Outcomes:
It is expected that students will gain a deep understanding of the fundamentals of digital electronics and acquire skills in designing and analyzing digital circuits and their practical applications.

- Recommended Textbooks:
- "Digital Design" by M. Morris Mano
- "Digital Electronics: Principles and Applications" by Roger L. Tokheim
- "Digital Electronics: A Practical Approach" by William Kleitz

- Prerequisites:
Students are recommended to have prior knowledge of basic electronics concepts, mathematics, and physics.

- Laboratory Topics:
- Introduction to Digital Electronics: Binary Number System, Boolean Algebra, and Logic Gates
- Practical experiments using basic logic gates: AND, OR, NOT, XOR, etc.
- Design and testing of simple combinational logic circuits.

- Multiplexers and Demultiplexers: Implementation and Applications.
- Design and testing of circuits using multiplexers and demultiplexers.
- Introduction to logic design tools (Logic simulation, etc.).
- Flip-Flops and Latches: Implementation and Applications.

Course Code: CCIT061
Course Title: object-oriented programming 1
Credit Units : 4-3-0-2

- Course Definition

An introduction to object-oriented programming principles using Python. Students will learn the basic concepts of object-oriented programming and how to apply them to problem-solving and application development using Python.

- Course Topics:

- Review of Programming Language: Introduction, Variables, Comments, Python Data Types
- Review of Programming Language: Operators, Conditions and if Statements, Loops
- Review of Programming Language: Functions, Arrays
- Introduction to Object Basics
- Closer Look at Class Members
- Constructors and Destructors
- Creating Functions within a Class (Lambda)
- Arrays of Objects - Part 1
- Arrays of Objects - Part 2
- Pointers to Objects
- Friend Functions
- Constructor Overloading
- Passing Objects (Classes) to Functions
- Returning Objects (Classes) from Functions
- Additional Examples

- Course Description:

This course aims to provide students with a basic understanding of object-oriented programming concepts and principles, as well as learning how to apply these concepts in writing programs and solving problems using Python.

- Learning Outcomes:
Students are expected to acquire the necessary skills to analyze problems, design programs, and write code using object-oriented concepts using Python, in addition to gaining a deep understanding of object-oriented programming fundamentals.

- Recommended Textbooks:
 - "Python Programming: An Introduction to Computer Science" by John Zelle
 or
 - "Object-Oriented Programming in Python" by Michael H. Goldwasser and David Letscher.

- Prerequisites:
Students are recommended to have a basic knowledge of programming, control structures, and language fundamentals, and they should be able to solve problems using a computer and think logically using Python.

- Laboratory Topics:
 - Introduction to Object-Oriented Programming (OOP) Concepts: Classes, Objects, and Encapsulation
 - Implementing Simple Classes and Objects in the Chosen Programming Language
 - Writing and Executing Object-Oriented Programs
 - Inheritance and Polymorphism: Extending Classes and Overriding Methods
 - Implementing Inheritance and Polymorphism in Object-Oriented Programs
 - Principles of Object-Oriented Design: Abstraction, Encapsulation, and Polymorphism
 - Applying Design Principles in Object-Oriented Program Development.

Course Code: CNDC209
Course Title: Algorithms
Credit Units : 3-2-0-2

- Course Definition
This course focuses on the study of algorithms, which are logical methods for solving problems using computers. It includes the development and

analysis of algorithms to solve a variety of computational and logistical problems.

- Course Topics:
  - Basic Concepts in Algorithm Analysis
  - Introduction to Algorithms
  - Big-O Notation
  - Linear Search Problem
  - Binary Search Problem
  - Sorting and Searching: Objective of Sorting, Sorting Steps
  - Bubble Sort
  - Heap Sort
  - Quick Sort, Merge Sort
  - Insertion Sort
  - Selection Sort
  - Graph Algorithms
  - Graph Search
  - Depth-First Search
  - Shortest Path Algorithm

- Course Description:

This course aims to introduce students to algorithmic concepts and teach them how to analyze algorithm performance and design them effectively to solve various computer problems.

- Learning Outcomes:

Students are expected to gain a deep understanding of algorithms and the ability to analyze and design algorithms to solve a variety of computer problems. They are also expected to acquire skills in evaluating algorithm performance and applying effective design principles.

- Recommended Textbooks:
  - "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein
  - "Algorithms" by Robert Sedgewick and Kevin Wayne.

- Prerequisites:

Students are recommended to have a basic knowledge of programming principles and mathematics. They are encouraged to develop analytical and logical thinking skills to deal with the complex computer challenges they will encounter in the course.

- Laboratory Topics:

- Introduction to Algorithm Analysis: Time Complexity and Space Complexity, Big-O Notation
- Implementation and Analysis of Basic Algorithms such as Linear Search, Binary Search, and Sorting Algorithms (e.g., Bubble Sort, Insertion Sort)
- Recursion and Recursive Algorithms: Understanding Recursive Problem-Solving Techniques
- Implementation and Analysis of Recursive Algorithms for Problems such as Factorial, Fibonacci Series, and Binary Search using Recursion
- Dynamic Programming: Understanding the Dynamic Programming Approach to Problem Solving
- Implementation and Analysis of Dynamic Programming Algorithms for Problems such as Knapsack Problem and Longest Common Subsequence.

Course Code: CNDC 207
Course Title: Computer Networks
Credit Units : 3-2-0-2

- Course Definition

This course introduces students to the concepts and techniques of computer networks, including network structure, protocols used, network configuration methods, network security, and network management.

- Course Topics:
  - Introduction to Computer Networks and TCP/IP Model
  - Switching: Three Switching Methods, Switching and TCP/IP Layers
  - Circuit Switched Networks: Three Stages, Efficiency, Delay
  - Packet Switching: Packet Networks, Virtual Circuit Networks
  - Data Link Layer Services and Addresses: Contracts and Links, Link Classes, Sublayers
  - Three Types of Addresses, Address Resolution Protocol (ARP)
  - Error Detection and Correction: Types of Errors, Redundancy, Detection vs. Correction
  - Data Link Control (DLC) Services: Frame Formation, Flow Control and Error Control
  - Data Link Control (DLC) Services: Unacknowledged Connectionless Services
  - Media Access Control (MAC) Part 1: Random Access: ALOHA, CSMA, CSMA/CD
  - Media Access Control (MAC) Part 2: Controlled Access
  - Network Layer Design Issues

- Routing, IPv4, IPv6

- Course Description:
This course covers various topics related to computer networks, including designing, configuring, securing, analyzing, managing, and troubleshooting common network problems.

- Learning Outcomes:
Upon completion of the course, students are expected to have a deep understanding of computer network concepts and the ability to design, manage, and secure networks, as well as skills in analyzing and solving related problems.

- Recommended Textbooks:
  - "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross
  - "Computer Networks" by Andrew S. Tanenbaum.

- Prerequisites:
Students are recommended to have a basic understanding of computing and networking concepts. Basic programming skills and understanding of operating systems and fundamental network protocols are preferred.

- Laboratory Topics:
  - Introduction to Computer Networks: Overview of Network Components, Topologies, Protocols
  - Setting Up a Basic Network: Configuring IP Addresses, Subnetting, Device Connectivity
  - Analysis of Network Packets: Using Tools like Wireshark to Capture and Analyze Network Traffic
  - Understanding the TCP/IP Protocol Suite and its Layers: Network Layer, Transport Layer, Application Layer
  - Configuring and Troubleshooting Local Area Networks (LANs): Setting Up Switches, VLANs, Spanning Tree Protocol
  - Implementing Basic Network Security Measures such as Access Control Lists (ACLs) and Port Security
  - Configuring and Troubleshooting Wireless Networks: Setting Up Wi-Fi Access Points and Security Protocols.

Course Code: CNDC206
Course Title: Microprocessors
Credit Units : 3-2-0-2

- Course Definition

This course introduces students to the concepts and techniques of microprocessors, including the structure and function of microprocessors, design principles, and programming skills.

- Course Topics:
  - Introduction to Microprocessors
  - 8086 Microprocessor Architecture
  - Registers, Memory Segmentation, Addressing Modes
  - Flags Register
  - Operating Modes: Minimum and Maximum Modes

- Course Description:

This course aims to familiarize students with microprocessor concepts and techniques, including the structure, function, design principles, and programming skills associated with microprocessors.

- Learning Outcomes:

After completing the course, students are expected to have a deep understanding of microprocessor concepts and the ability to design, program, and troubleshoot microprocessor-based systems.

- Recommended Textbook:
  - "Introduction to the Theory of Computation" by Michael Sipser.

- Prerequisites:

Students are recommended to have a basic understanding of computing concepts and digital systems. Basic programming skills may be beneficial but are not required.

- Laboratory Topics:
  - Introduction to Microprocessors
  - Microprocessor Programming
  - Assembly Language Programming
  - Hardware Interfacing and Peripheral Devices.

- Arithmetic and Logical Operations: Addition, subtraction, multiplication, division, and logical operations.

- Writing Assembly Language Programs for Arithmetic and Logical Operations.
- Debugging and Testing Assembly Language Programs.
- Control Flow Instructions: Conditional branching, loops, and subroutines.

Course Code: CCIT062
Course Title: Numerical Analysis
Credit Units : 3-2-0-2

- Course Definition

This course involves the study and application of mathematical and computational methods and techniques in solving various numerical and mathematical problems. Topics covered include root approximation, integration calculation, differential equation solving, error estimation, and more.

- Course Topics:
  - Direct Methods for Solving Systems of Linear Equations
  - Simple Gaussian Elimination Method
  - Gaussian Elimination with Partial Pivoting
  - Determinants Estimation
  - Gauss-Jordan Method
  - LU Decomposition Analysis: Doolittle Decomposition
  - LU Decomposition Analysis: Doolittle Method with Row Swapping
  - Finding the Inverse Matrix
  - Iterative Methods for Solving Systems of Linear Equations
  - Jacobi Iteration Method
  - Gauss-Seidel Method
  - Successive Over-Relaxation Method (SOR Method)
  - Newton-Raphson Method
  - Runge-Kutta Method
  - Numerical Differentiation and Integration
  - Interpolation and Lagrange Multipliers
  - Numerical Approximation and Neville's Method
  - Numerical Analysis Methods for Differential Equations
  - Numerical Analysis Methods for Integral Equations

- Course Description:

The course focuses on teaching students numerical solution techniques for various mathematical problems using computation, along with understanding the theory and practical applications of these techniques.

- Learning Outcomes:
Upon completion of the course, students are expected to acquire skills in using numerical analysis methods to solve a variety of mathematical and engineering problems. Additionally, they should be able to evaluate solution accuracy and analyze errors.

- Recommended Textbooks:
"Numerical Analysis" by Richard L. Burden and J. Douglas Faires, and "Numerical Methods for Engineers" by Steven C. Chapra.

- Prerequisites:
It is preferable for students to have prior knowledge of basic mathematics, including algebra, differentiation, and integration, along with skills in programming using the preferred computer language.

- Lab Content:
- Introduction to Numerical Analysis: Overview of numerical methods and their applications.
- Implementation and analysis of basic numerical algorithms such as numerical integration using the trapezoidal rule and Simpson's rule.
- Solving systems of linear equations: Introduction to direct and iterative methods.
- Implementation and analysis of direct methods such as Gaussian elimination and LU decomposition.
- Function approximation: Techniques of interpolation and curve fitting with data.
- Implementation and analysis of interpolation methods such as Lagrange interpolation and spline interpolation.
- Numerical differentiation and integration: Techniques for computing derivatives and integrals numerically.

- Course Code (CCIT063):
- Course Title (Object-Oriented Programming 2):
Credit Units 4-3-0-2)

- Course Definition:
This course aims to teach students the concepts of object-oriented programming and apply them using the Python programming language. It includes studying the fundamental principles of object-oriented programming and its applications in software development.

- Course Topics:
  - Introduction to Operator Overloading
  - Operator Overloading Using Member Functions
  - Access Control in Base Class
  - Introduction to Inheritance
  - Inheritance from Multiple Base Classes
  - Constructors, Destructors, and Inheritance
  - Passing Arguments to Base Class Constructors
  - Using Public, Protected, and Private Members in the Base Class
  - Method Overriding in Python Inheritance
  - Examples of Inheritance
  - Composition
  - Multilevel Inheritance
  - Hierarchical and Hybrid Inheritance

- Course Description:
  The course focuses on teaching students how to design and implement programs using the principles of object-oriented programming in the Python programming language. It covers topics such as object creation, inheritance creation and management, and the use of aggregates and functions.

- Learning Outcomes:
  Upon completion of the course, students are expected to understand and apply the concepts of object-oriented programming in the Python programming language, and to design and develop software using these concepts.

- Recommended Textbooks:
  "Python 3 Object-Oriented Programming" by Dusty Phillips, and "Learning Python" by Mark Lutz.

- Prerequisites:
  It is preferred for students to have prior knowledge of the Python programming language and basic programming concepts.

- Lab Topics:
  - Review of Object-Oriented Programming Concepts: Revisiting fundamental concepts such as classes, objects, inheritance, polymorphism, and encapsulation.
  - Advanced Inheritance: In-depth study of inheritance, including multiple inheritance, default inheritance, and the diamond problem.

- Templates and Generic Programming: Understanding templates and their use in creating generic classes and functions.
- Exception Handling: Introduction to exception handling mechanisms for dealing with errors during runtime and exceptional cases.
- Advanced Object-Oriented Design Patterns: Study and application of advanced design patterns such as singleton, factory, observer, and adapter.
- Advanced File Handling: Techniques for reading and writing files, serialization, and handling binary files.
- Advanced Memory Management: Concepts such as smart pointers, garbage collection, and memory profiling for efficient memory management.

- Course Code: (CNDC213)
- Course Title: (Web Design)
- Credit Units (3-2-0-2)

- Course Definition:
The Web Design course aims to introduce students to the concepts of designing and developing websites.

- Course Topics:
- Introduction to Web Design:
History and evolution of web design
Understanding the principles of web design, including layout, typography, color theory, and user experience (UX)
- HTML and CSS:
Introduction to HTML tags and syntax
Understanding CSS selectors and properties
Creating simple HTML pages and applying CSS styles
- HTML Elements, HTML Lists, HTML Links, HTML Images, Tables, Frame Tags, and Their Attributes
- Web Development Tools and Frameworks:
Understanding web development tools such as text editors, version control systems, and package managers
Familiarity with popular front-end frameworks like Bootstrap and React
Understanding how to integrate external APIs and web application interfaces
- Introduction to JavaScript:
Declaring JavaScript variables and performing arithmetic operations

Adding strings and numbers, comparison, and logical operations in JavaScript
Introduction to conditional statements and dialog boxes in JavaScript
- Graphics and Multimedia on the Web:
Using graphic design tools to create web graphics
Understanding different file formats (e.g., JPEG, PNG, SVG)
Integrating multimedia elements such as video and audio.
- HTML and CSS:
- Introduction to HTML tags and their syntax
- Understanding CSS styles and selectors
- Creating simple HTML pages and applying CSS formatting
- HTML Elements, HTML Lists, HTML Links, HTML Images, Tables, Frame Tags, and Their Attributes
- Web Development Tools and Frameworks:
- Understanding web development tools such as text editors, version control systems, and package managers
- Familiarity with popular front-end frameworks like Bootstrap and React
- Understanding how to integrate external APIs and web application interfaces
- Introduction to JavaScript:
- Declaring JavaScript variables and performing arithmetic operations
- Adding strings and numbers, comparison, and logical operations in JavaScript
- Introduction to conditional statements and dialog boxes in JavaScript
- Graphics and Multimedia on the Web:
- Using graphic design tools to create web graphics
- Understanding different file formats (e.g., JPEG, PNG, SVG)
- Integrating multimedia elements such as video and audio.

- Course Description:
This course covers the design and development of websites using multiple languages and technologies, with a focus on user experience and best design practices.

- Learning Outcomes:
Students are expected to acquire skills in designing and implementing websites professionally, as well as understanding the concepts of interface design and user experience.

- Recommended Textbooks:
1. "HTML and CSS: Design and Build Websites"
2. "JavaScript and jQuery: Interactive Front-End Web Development"

3. "Don't Make Me Think: A Common Sense Approach to Web Usability"

- Prerequisites:
There are no specific prerequisites for this course, but it is preferable for students to have a basic understanding of computer fundamentals and the Internet.

- Lab Topics:
 - Principles of Web Design: Understanding the importance of user experience, accessibility, and visual design.
 - HTML Basics: Creating the structure of web pages using HTML tags.
 - CSS Basics: Formatting web pages using CSS properties and selectors.
 - Layout and Positioning: Creating responsive layouts and positioning elements on web pages.
 - Typography and Color Theory: Selecting appropriate fonts and colors for web design.
 - Image Optimization: Optimizing images for display on the web and integrating them into web pages.
 - Navigation Design: Creating intuitive and user-friendly navigation menus.

Course code (UOA011)
– Course title (English2)
- Credit Units(2_2_0_0)

- **Course Definition : Improve students' reading comprehension skills by exposing them to a variety of authentic texts, including literary works, academic articles, and news articles. Develop their ability to analyze and interpret texts, identify main ideas, and understand the author's purpose and tone.**

- **Course Topics:**

Getting to Know you; Grammar

Getting to Know you; Vocabulary; Everyday English

The Way We Live; Grammar

The Way We Live; Vocabulary; Everyday English

It All Went Wrong; Grammar

It All Went Wrong; Vocabulary; Everyday English

Let Us Go Shopping; Grammar

Let Us Go Shopping; Vocabulary; Everyday English

What Do You Want To Do; Grammar

What Do You Want To Do; Vocabulary; Everyday English

Tell Me What's it Like; Grammar

Tell Me What's it Like; Vocabulary; Everyday English

Famous Couples; Grammar

Famous Couples; Vocabulary; Everyday English

Do's and Don'ts; Grammar; Vocabulary; Everyday English

- **Course Description : Improve students' reading comprehension skills by exposing them to a variety of authentic texts, including literary works, academic articles, and news articles. Develop their ability to analyze and interpret texts, identify main ideas, and understand the author's purpose and tone.**

- **Course Outcomes: Demonstrate improved reading comprehension skills by understanding and analyzing a variety of authentic texts, including literary works, academic articles, and news articles. Identify main ideas, supporting details, and the author's purpose and tone.**

- **Recommended Textbook(s): New Head Way Pre-Intermediate Level; Liz and John Soars; OXFORD**

Course Code: CNDC309
Course Title: Database Management Systems (DBMSs)
Credit Units: 3-2-0-2

Course Definition:
The course studies database management systems and their applications in efficiently and effectively storing and retrieving data.

Course Topics:
- Introduction to Database Management Systems
- Overview of Data, Data Abstraction, Instances, and Schemas
- Designing Distributed Databases
- Database Languages: Definition and Manipulation Language, Query Language
- Conceptual Database Design - Entity-Relationship (ER) Modeling
- Relational Model, Types of Keys
- Relational Algebra
- Relational Calculus, Tuple Relational Calculus, Examples
- Hierarchical Query Language (SQL)
- Domain Relational Calculus, Examples of DRC Queries
- SQL, Basic SQL Query Format + Examples (1)
- SQL, Basic SQL Query Format + Examples (2)
- Schema Refinement
- Normalization
- Functional Dependencies, Decomposition

Course Description:

The course focuses on studying the design, development, and management of database systems along with their practical applications in projects.

Course Learning Outcomes:
Students are expected to gain a deep understanding of database management systems and the ability to apply these concepts in building database applications.

Recommended Textbook(s):
- "Database System Concepts" by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan.
- "Fundamentals of Database Systems" by Ramez Elmasri and Shamkant B. Navathe.

- Prerequisites:
There are no prerequisites, but it is preferable for students to have a basic understanding of programming languages and software design.

- Lab Topics:
- Relational Database Design: Principles and techniques of relational database design, including Entity-Relationship modeling and normalization.
- SQL Fundamentals: Introduction to Structured Query Language (SQL) for data manipulation, querying, and database management.
- Database Implementation: Hands-on experience in creating and managing databases using popular database management systems such as MySQL or Oracle.
- Advanced SQL Concepts: Advanced features and techniques of SQL for complex queries, joins, subqueries, and transactions.
- Database Management: User access management, database security, backup and recovery, and performance tuning.
- Project Development: Application of concepts learned throughout the course to develop a database application or project.

Course Code: CNDC308
Course Title: Digital Signal Processing 1
Credit Units: 2-2-0-0

- Course Definition:
The course studies digital signal processing, its analysis, and the design of its applications.

- Course Topics:
- Introduction to Digital Signal Processing
- Signal Sampling and Quantization
- Digital-to-Analog Conversion
- Digital Signals and Systems
- Filter Design
- Linear Time-Invariant (LTI) Systems, Causal Systems
- Signal Manipulation
- Differential Equation Formulation
- Digital Recursive Convolution
- Applications of Digital Signal Processing
- Recursive Transformation Methods
- Fourier Transform
- Fourier Transform
- Digital Filters
- Adaptive Signal Processing

- Course Description:
The course focuses on studying the concepts of digital signal processing and its applications in various fields such as communications, audio, and medical.

- Course Learning Outcomes:
Students are expected to gain a deep understanding of digital signal analysis and processing and its practical applications.

- Recommended Textbook(s):
- "Digital Signal Processing: Principles, Algorithms, and Applications" by John G. Proakis and Dimitris K. Manolakis.
- "Understanding Digital Signal Processing" by Richard G. Lyons.

- Prerequisites:
It is preferable for students to have a basic understanding of mathematics and programming.

- Lab Topics:
None

Course Code: CCIT064
Course Title: Visual Programming I

Credit Hours: 3-2-0-2

- **Course Definition**:
The course focuses on software development using the C# language with an emphasis on visual programming applications.

- **Course Topics**:
  - Basics of C# Programming: Control Structures (if-else, loops)
  - Arrays
  - One-Dimensional Arrays
  - Multidimensional Arrays
  - Basics of C# Programming: Functions in C#
  - Function Overloading
  - Recursion in Functions
  - Introduction to Strings
  - Regular Expressions, Custom Structures (Struct), and Enumerations (Enum)
  - Object-Oriented Programming in C#: Introduction to object-oriented programming concepts in C#
  - Object-Oriented Programming in C#: Creating Classes, Objects, and Functions in C#
  - Object-Oriented Programming in C#: Creating Classes, Objects, and Functions in C#

- **Course Description**

The course aims to teach students programming concepts using the C# language, with a focus on visual programming applications such as desktop and web applications.

- **Learning Outcomes:**
It is expected that students will acquire skills in developing applications using C# and gain a deep understanding of visual programming applications.

- **Recommended Textbook(s):**
- "C# 9 and .NET 5 – Modern Cross-Platform Development" by Mark J. Price.
- "Programming C# 8.0: Build Windows, Web, and Desktop Applications" by Ian Griffiths.

- **Prerequisites:**

It is preferable for students to have a basic understanding of the C# programming language and visual programming.

- **Lab. Topics**:
  - Introduction to Visual Programming: Overview of visual programming concepts, its benefits, and applications.
  - User Interface Design: Designing user-friendly and attractive user interfaces using drag-and-drop tools and components.
  - Event Handling: Implementing event-driven programming using visual programming tools to respond to user interactions.
  - Data Manipulation: Working with data structures such as arrays and lists and performing operations such as sorting and filtering.
  - Graphics and Multimedia Elements: Integrating graphics, images, and multimedia components into visual applications.
  - Database Connectivity: Connecting visual applications to databases, retrieving data, and managing database records.
  - Project Development: Applying the concepts learned throughout the course to develop a complete project in visual programming.

Course Code: CNDC303
Course Title: Web Programming
Credit Hours: 3-2-0-2

- **Course Definition**

Study of developing web applications using programming languages such as HTML, CSS, and JavaScript, in addition to popular frameworks like Django or Flask.

- **Course Topics:**
  - Principles of PHP:
    - Introduction to PHP and scripting language concepts.
    - PHP syntax and structure.
    - Using PHP and its significance.
    - PHP file extensions.
  - Basics of PHP:
    - Data types in PHP, variables, constants, and operators.
- Comments in PHP.
- PHP Logic:
  - Control Structures in PHP: If-else, Switch Case.
  - Loops in PHP: For, Each, While, Do While.
  - Strings in PHP and string functions.

- Functions in PHP: Built-in, string, numerical.

- PHP Development:
  - PHP Functions and Objects.
  - Regular expressions in PHP.
  - Login form in PHP using GET and POST methods.
  - PHP sessions and cookies.
  - File handling in PHP.
  - Sending email using the mail() function in PHP.
  - Using MySQLi functions in PHP.

- Course Description:

The course aims to teach students how to develop web applications from scratch using core web languages and established frameworks.

- Course Outcomes:

Students are expected to acquire skills in designing and developing web applications with a deep understanding of web programming fundamentals.

- Recommended Textbooks:

- "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins.
- "Eloquent JavaScript: A Modern Introduction to Programming" by Marijn Haverbeke.
- "Flask Web Development: Developing Web Applications with Python" by Miguel Grinberg.

- Prerequisites:

It's preferable for students to have a basic understanding of programming languages such as Python and JavaScript.

- **Lab. Topics**

  - Overview of web development technologies and client-server engineering.
  - Creating and styling web pages using HTML markup and CSS stylesheets.
  - Introduction to JavaScript programming language for implementing interactivity and dynamic behavior on web pages.

- Introduction to server-side programming languages such as PHP, Python, or Node.js for handling form submissions and server-side processing.
- Connecting web applications to databases, querying and manipulating data using SQL, and implementing CRUD operations (Create, Read, Update, Delete).
- Introduction to popular frameworks for rapid web application development such as Django, Ruby on Rails, or ASP.NET.
- Deploying web applications on web servers, configuring hosting environments, and ensuring security measures.

Course Code:CNDC306
Course Title: Wireless Networks
Credit Units: 3-2-0-2

- **Course Definition**

This course covers the design, operation, and management of wireless networks, including WLANs, WWANs, and WPANs.

- **Course Topics**

- Introduction to wireless communications.
- Definition and overview of wireless communication technologies.
- Wireless communication technology review.
- Point-to-point communication across wireless networks: basic principles, communication division.
- Configuration of layers and end-to-end argument.
- Basic Transmission Control Protocol (TCP).
- TCP over wireless networks.
- Multiple access with time division, frequency, and code (FDMA, TDMA, CDMA).
- Carrier Sense Multiple Access with Collision Detection (CSMA/CD).
- Layer I: MACA and MACAW-1.
- Layer II: MACA and MACAW-2.
- Wireless mesh networks.
- Diversity routing in wireless networks.
- Cellular wireless networks.

- Wireless sensor networks.
- Bit error detection and correction.

- **Course Description**

The course aims to teach students how to design, operate, and manage wireless networks, including understanding the technologies used, security challenges, and best practices.

- **Course Outcomes**

Students are expected to gain a deep understanding of wireless networks and their ability to design, operate, and manage wireless networks effectively.

- **Recommended Textbook(s):**

- "Wireless Communications & Networks" by William Stallings.
- "Wireless Networking Complete" by Pei Zheng.
- "802.11 Wireless Networks: The Definitive Guide" by Matthew Gast.

- **Prerequisites:**

It is preferable for students to have a basic understanding of fundamentals of both wired and wireless networks.

- **Lab. Topics**

  - Introduction to wireless networks: Overview of wireless communication, network engineering, and protocols.
  - WLANs: Understanding Wi-Fi technology, IEEE 802.11 standards, and configuring wireless access points.
  - Wireless network security: Implementing security measures such as encryption, authentication, and intrusion detection in wireless networks.
  - Wireless sensor networks: Exploring principles and applications of wireless sensor networks and conducting practical experiments.
  - **Random Mobile Networks:** Understanding concepts and protocols of random mobile networks and simulating scenarios of random network formations.

  - **Wireless Network Performance Analysis:** Analyzing and evaluating the performance of wireless networks using tools and techniques.

- **Project Development:** Developing a solution for a small-scale wireless network or conducting a research project related to wireless networks.

Course Code:CNDC313
Course Title: Digital Signal Processing II
Credit Units: 2-2-0-0

- **Course Definition**

This course focuses on advanced concepts in digital signal processing, including the analysis of multi-dimensional signals and the design and implementation of digital systems.

- **Course Topics**

- Discrete Fourier Transform (DFT)
- Inverse Discrete Fourier Transform (IDFT)
- Amplitude and Power Spectrum (ASPS)
- Fast Fourier Transform (FFT)
- Multi-rate Digital Signal Processing (MDSP)
- Fast Fourier Transform - Frequency Domain Decimation Method
- Time Domain Decimation Method
- Differential Equation and Digital Signal Filtering
- Basic Filtering Types
- Implementation of Digital Signal Processing
- Principles and Algorithms of Digital Signal Processing
- Realization of Digital Filters
- Application: Speech Enhancement and Sound Filtering
- Finite Impulse Response Filter Design
- Windowing Method (Noise Function or Attenuation Function)

- **Course Description**

This course aims to expand knowledge in the field of digital signal processing by studying advanced concepts and their practical applications.

- **Course Outcomes**

Students are expected to acquire skills in analyzing, designing, and implementing digital signal processing systems with a deep understanding of their diverse applications.

- **Recommended Textbook(s):**

- "Digital Signal Processing: Principles, Algorithms, and Applications" by John G. Proakis and Dimitris G. Manolakis.
- "Digital Signal Processing: A Practical Guide for Engineers and Scientists" by Steven W. Smith.

- **Prerequisites:**

It is preferable for students to have a basic understanding of digital signal processing concepts.

- **Lab. Topics**

None.

Course Code:CNDC322
Course Title: Distributed Database Management Systems

Credit Units: 3-2-0-2

- **Course Definition**

This course studies the design, implementation, and management of distributed databases that span multiple geographic locations.

- **Course Topics**

- Introduction to Distributed Databases (DDB)
- Functions of Database Management Systems (DBMS)
- Responsibilities of Database Administrators (DBA)
- Distributed Database Facilities (DDB)

- Constraints of Distributed Databases (DDB)
- Advantages and disadvantages of Distributed Databases (DDB)
- Architectural engineering of Distributed Database (DDB) and components of Distributed Database Management Systems (DDBMS)
- Overview of Distributed Databases (DDB) and Distributed Database Management Systems (DDBMS)
- Levels of data and operations distribution
- Safety of Distributed Databases (DDB)
- Transparency features of Distributed Databases
- Query processing and optimization
- Query scenarios
- Transaction transparency
- DO-UNDO-REDO protocol
- Design of Distributed Databases
- Replication and data partitioning
- Data retrieval
- Data storage and retrieval

- **Course Description**

The course aims to teach students how to design, implement, and manage Distributed Databases while understanding the related challenges and strategies.

- **Course Outcomes**

Students are expected to gain skills in designing and managing Distributed Databases and understand the technical and practical challenges.

- **Recommended Textbook(s):**

- "Distributed Database Systems" by Stefano Ceri and Giuseppe Pelagatti.
- "Principles of Distributed Database Systems" by M. Tamer Özsu and Patrick Valduriez.

- **Prerequisites:**

Students are encouraged to have basic knowledge of databases and information systems.

- - **Lab. Topics**

- Introduction to Distributed Database Management Systems (DDBMS): An overview of distributed database concepts, architecture, and advantages.
- Designing Distributed Databases: Techniques for designing distributed databases, including data partitioning, replication, and retrieval.
- Query Processing and Optimization: Understanding query processing and optimization in a distributed environment, including parallel processing techniques.
- Transaction Management: Dealing with distributed transactions, monitoring synchronization, deadlock detection, and resolution in a distributed environment.
- Data Replication and Consistency: Exploring data replication techniques, consistency models, and consistency preservation protocols in a distributed database.
- Security of Distributed Databases: Security measures and protocols for protecting data in a distributed database system.
- Management of Distributed Databases: Practical aspects of managing distributed databases, including backup and recovery, performance tuning, and monitoring.

Course Code CNDC302
Course Title Multimedia
- Credit Units - Lecture Hours - Practical Hours - Lab Hours (3-2-0-2)

- **Course Definition**

Study of designing, developing, and analyzing multimedia such as audio, image, and video and their applications in interactive media.

- **Course Topics**

- Introduction to Multimedia
- Basics of Multimedia Data
- Multimedia Elements and Tools
- Graphics and Image Editing
- Image Compression
- Audio and Sound Editing
- Video Editing
- Animation and Interactive Media
- Multimedia and Web Development

- Multimedia and Social Media
- Multimedia and Marketing
- Multimedia and Gaming
- Multimedia and Cinema
- Multimedia and Education
- Ethics and Legal Issues in Multimedia

- **Course Description**

The course aims to teach students how to apply concepts and techniques of multimedia in designing and developing innovative interactive applications.

- **Course Outcomes**

Students are expected to acquire skills in designing, implementing, and analyzing multimedia applications and understanding the technologies used in them.

- **Recommended Textbook(s):**

- "Multimedia: Making It Work" by Tay Vaughan.
- "Multimedia Systems" by Ralf Steinmetz and Klara Nahrstedt.

- **Prerequisites:**

Students are advised to have a basic understanding of programming and multimedia technologies.

- **Lab. Topics**

  - Introduction to Multimedia: An overview of multimedia concepts, components, and applications.
  - Digital Imaging: Image manipulation techniques, image formats, and editing tools.
  - Audio Processing: Recording, editing, and playback techniques. Introduction to audio formats and compression.
  - Video Processing: Video capture, editing, and encoding. Understanding video formats and compression algorithms.
  - Multimedia Authoring Tools: Hands-on experience with multimedia authoring software to create interactive multimedia presentations.

- Multimedia Programming: An introduction to multimedia programming languages and frameworks for creating interactive multimedia applications.
- Multimedia Applications: Exploring practical applications of multimedia in fields such as education, entertainment, advertising, and virtual reality.

Course Code (CNDC307)
Course Title (Network Programming:)
Units - Lecture Hours - Practical Hours - Lab Hours (3-2-0-2)

- - **Course Definition**:

Study of designing and developing network applications using network programming and communication protocols.

- - **Course Topics**

• Introduction
  - Brief history of networks
  - Motivations and consequences
  - Features and scope of network programming
  - Overview of network programming in Python
• Basics of Networks and the Web
  - Network, hosts, and addresses
  - Types of networks
  - Internet and World Wide Web
  - Network models and layers
  - OSI reference model
  - Network protocols
  - Network standards
• Intensive Python Course
  - Introduction to Python
  - Python data types
  - Working with lists
  - Input/output dictionaries
  - Functions
- Classes and Object-Oriented Programming
- Files and Exceptions
• Overview of Network Programming in Python
  - Network Programming Support in Python
  - Network Programming Libraries in Python
• Addresses, Naming, and Domain Name System (DNS)

- Working with IPv4 Addresses
- Working with Domain Names
- Working with IPv6 Addresses

• Socket Programming
- Socket Concepts
- Sending/Receiving Data over Sockets
- Buffer Size and Timeout
- Blocking/Non-blocking Mode

• TCP Programming
- TCP Concepts
- TCP Protocol and Message Format
- Simple Application for TCP Echo Client-Server

• UDP Programming
- UDP Concepts
- UDP Protocol and Message Format
- Simple Application for UDP Echo Client-Server

• Graphical User Interface (GUI) Programming in Python
- GUI Frameworks in Python
- Tkinter, wxPython, Kivy, PyQt
- GUI and Networking in Python

• HTTP Programming for the Internet
- HTTP Protocol
- Sending/Receiving HTTP Requests/Responses
- Making HTTP Requests and Setting/Sending HTTP Responses
- Dealing with Forms
- Handling Cookie Information

• Email Processing
- Email Protocols and Handling
- SMTP (Simple Mail Transfer Protocol) Programming
- POP3 (Post Office Protocol - Version 3) Programming
- IMAP (Internet Message Access Protocol) Programming
- Working with Gmail from Google

• Cross-Machine Programming
- Telnet and Remote Access
- FTP and SFTP

- **- Course Description**

The course aims to teach students how to apply networking concepts in designing and developing various network applications.

- **- Course Outcomes**

Students are expected to gain skills in programming network applications and understanding the related challenges and strategies.

- - **Recommended Textbook(s):**

- "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross.
- "TCP/IP Sockets in C: Practical Guide for Programmers" by Michael J. Donahoo and Kenneth L. Calvert.

- - **Prerequisites:**

Students are encouraged to have a basic understanding of programming and networking concepts.

- **Lab. Topics**

  - Introduction to Network Programming: An overview of network protocols, socket programming, and client-server architecture.
  - Network Communication: Understanding the basics of network communication, including TCP/IP and UDP protocols.
  - Socket Programming: Practical experience in socket programming using a programming language like Python or Java.
  - Network Protocols: Exploring common network protocols such as HTTP, FTP, and SMTP, and implementing client and server applications using these protocols.
  - Network Security: Introduction to network security principles, including encryption, authentication, and secure communication protocols.
  - Network Applications: Developing network applications such as chat systems, file transfer applications, and remote procedure calls.
  - Network Troubleshooting and Performance Optimization: Techniques for troubleshooting network issues and improving network performance.

- Course Code (CNDC301)
Course Title (Software Engineering:)
- Credit Units - Lecture Hours - Practical Hours - Lab Hours (2-2-0-0)

-
- **Course Definition**

Study of software engineering concepts, software development methodologies, and software project management techniques.

- **- Course Topics**

  - Introduction to Software Engineering and Software Applications:
  - Defining software engineering, exploring the evolving role of software, and discussing software characteristics and principles of software engineering.
  - Characteristics of Software Engineers and Software Applications:
  - Understanding the characteristics of software engineers, developing software applications, and discussing potential crises in the field.
  - Software Engineering Models:
  - Exploring progressive techniques in software engineering, and software engineering models such as the waterfall model, prototyping model, evolutionary software process model, incremental model, spiral model, and agile model.

  - Introduction to Software Process and Project Metrics:
  - Understanding the software process and project metrics, including measurements and indicators, process metrics, metrics in practice, project domains, project metrics, and software measurement.
  - Software Quality Metrics and Integration with Software Process:
  - Exploring size-oriented metrics, function-oriented metrics, function point calculation, defect removal efficiency, and metrics for integration with the software process.

- Course Description:

The course aims to introduce students to software engineering concepts and teach them how to apply them in developing high-quality and maintainable software.

- Course Outcomes

Students are expected to gain skills in applying software development methodologies and software project management techniques.

- **Recommended Textbook(s):**

  - "Software Engineering: A Practitioner's Approach" by Roger S. Pressman.
  - "Agile Estimating and Planning" by Mike Cohn.

- Prerequisites:

It is preferable for students to have knowledge of programming fundamentals and software engineering basics.

- - Lab. Topics

None

- Course Code (CCIT065)
Course Title (Visual Programming II:)
- Credit Units - Lecture Hours - Practical Hours - Lab Hours (3-2-0-2)

- Course Definition

Study of C# programming language applications in developing desktop and web applications using appropriate Integrated Development Environment (IDE).

- Course Topics

- Windows Forms: Introduction to Windows Forms for developing graphical user interfaces in C#.
- C# Windows Forms: Creating simple GUI applications using Windows Forms in C#.
- C# Windows Forms: Model (Properties, Methods, and Events) in Windows Forms in C#.
- C# Windows Forms: Control in Windows Forms.
- Advanced User Interface Enhancement: Dialog Boxes.
- Advanced User Interface Enhancement: Types of Dialog Boxes.
- Modifying Form Properties.
- Error Handling in Winforms: Exceptions, Error Types.
- Error Handling in Winforms: Exception Classes, Exception Properties, Exception Handling, ErrorProvider Control Interface.
- Introduction to ADO.Net Data Access Components.

- Advanced Windows Forms Programming: Developing more complex GUI applications using Windows Forms and C#.
- Advanced Windows Forms Programming: Understanding event-driven programming and handling user input.
- Web Development with C#: Introduction to web development using C#.
- Web Development with C#: Creating simple web applications using ASP.NET and C#.
- Creating Windows Model Projects.

- Course Description:

The course aims to teach students how to effectively use the C# programming language in developing advanced desktop and web applications.

- - Course Outcomes

Students are expected to acquire skills in advanced C# application development and understand the technologies used in desktop and web applications.

- Recommended Textbook(s):

- "C# 9 and .NET 5 – Modern Cross-Platform Development" by Mark J. Price.
- "Pro ASP.NET MVC 5" by Adam Freeman.

- Prerequisites:

Students are recommended to have basic knowledge of the C# programming language and application development.

- Lab. Topics

- Advanced Graphical User Interface Development: Building interactive and user-friendly graphical user interfaces using visual programming tools.
- Event Handling: Implementing event-driven programming techniques to respond to user actions and trigger appropriate functions or operations.
- Data Binding: Connecting data models to user interface elements to achieve synchronization and dynamic interface updates.
- Advanced Controls: Exploring advanced user interface controls and components such as menus, toolbars, tab controls, and grids.

- Integration with Multimedia: Integrating multimedia elements such as images, audio, and video into graphical user interfaces.
- File Handling: Working with files and folders, performing file operations such as reading, writing, and file management.
- Error Handling and Debugging: Implementing error handling mechanisms and debugging techniques to identify and fix issues in visual programming code.

Course Code: CNDC401
Course Title: Operating Systems
Credit Hours: 3-2-0-2

- Course Description:
The course covers the study of operating systems, their functions, and how they interact with hardware and other software components.

- Course Topics:
- Overview of Operating Systems
- Processes and Functions in Operating Systems
- Structures in Operating Systems
- Process Description and Control
- Threading
- Synchronization
- CPU Scheduling
- Deadlocks and Starvation - Part 1
- Deadlocks and Starvation - Part 2
- Memory Management: Swapping
- Memory Management: Partitioning
- Memory Management: Virtual Memory
- Input/Output Management
- File Systems

- Course Learning Outcomes:

Students are expected to gain a comprehensive understanding of operating systems and their ability to apply these concepts in designing and developing software.

-Recommended Textbook(s):
- "Operating System Concepts" by Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne.
- "Modern Operating Systems" by Andrew S. Tanenbaum and Herbert Bos.

-Prerequisites:
It is preferable for students to have a basic understanding of programming languages and computer systems.

-Lab Topics:
- Introduction to Operating Systems: Overview of operating systems, their functions, and their role in computer systems.
- Process Management: Understanding process creation, scheduling, synchronization, and communication.
- Memory Management: Concepts and techniques for memory allocation, paging, and virtual memory.
- File Systems: Organization of file systems, file operations, directory structures, and file access control.
- Input/Output Systems: Managing input/output devices, device drivers, and input/output operations.
- Deadlocks: Understanding deadlock prevention, avoidance, detection, and recovery.
- Case Studies and Implementation: Studying and implementing specific components of the operating system, such as scheduling algorithms, memory management techniques, or file system operations.

Course Code: CCIT066
Course Title: Artificial Intelligence I
Credit Hours - Lecture Hours - Practical Hours - Lab Hours: 3-2-0-2

- **Course Definition**

The Artificial Intelligence I course is an essential part of computer science curricula, focusing on studying the concepts of artificial intelligence and its applications in areas such as machine learning and data analysis.

- **Course Topics**

- General Introduction.
- History of Artificial Intelligence.
- Systematic Search: Basic Graph Concepts; State Space Representation of Problems.
- Depth-First Search.
- Breadth-First Search.
- Hybrid Search.
- Propositional Logic and Resolution in Propositional Logic.
- Predicate Logic: Basic Concepts and Definitions.
- Predicate Logic: Examples.
- Horn Clauses.
- Unification and Skolemization.
- Clause Normal Form.
- Modus-Ponens and Resolution Inference Rules in Predicate Logic.
- Control Strategies for Resolution Inference (Problem Solving).

- **Course Description**

- Comprehensive understanding of artificial intelligence concepts.
- Ability to apply machine learning techniques to real-world problem-solving.
- Skills required to implement artificial intelligence projects.

- **Course Outcomes**

-Comprehensive understanding of artificial intelligence concepts.
-Ability to apply machine learning techniques in solving practical problems.
- Skills necessary to implement artificial intelligence projects.

-Recommended Textbooks:

- "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig.
- "Pattern Recognition and Machine Learning" by Christopher M. Bishop.

- Prerequisites:
There are no prerequisites.

- Lab Topics:

- Introduction to Artificial Intelligence: Overview of artificial intelligence concepts, its history, and applications.
- Problem Solving and Search: Implementing search algorithms such as depth-first search, breadth-first search, and A* search to solve problems.
- Knowledge Representation and Logic: Representing knowledge using propositional and first-order logic, and implementing inference algorithms.
- Machine Learning: Introduction to machine learning algorithms, including supervised learning, unsupervised learning, and reinforcement learning.
- Natural Language Processing: Implementing natural language processing techniques for tasks such as text classification, sentiment analysis, and language generation.
- Neural Networks: Introduction to neural networks and deep learning, implementing advanced neural networks and recurrent networks for classification tasks.
- Ethics of Artificial Intelligence and its Applications: Discussing ethical considerations in artificial intelligence and exploring practical applications of AI in various domains.

Course Code (CNDC413)
Course Title (Information Security)
Credit Hours - Lecture Hours - Practical Hours - Lab Hours (2-2-0-0)

- Course Definition:
The Information Security course deals with studying how to protect information and data from unauthorized access, manipulation, destruction, ensuring their integrity and confidentiality.

- Course Topics

- Introduction
- Information Security Models
- Classical Encryption Techniques Part I
- Statistical Attacks
- Classical Encryption Techniques Part II
- Block Ciphers

- Data Encryption Standard
- Security of Data Encryption Standard
- Mathematical Fundamentals
- Group Theory
- Rings and Fields
- Modular Arithmetic
- Prime Finite Fields
- Using Block Ciphers in Real-World Systems
- Modes of Operation

- Course Description:

The Information Security course aims to introduce students to the concepts of cybersecurity and its techniques, teaching them how to analyze threats, assess risks, and implement appropriate defense strategies.

- Course Outcomes

- Deep understanding of the principles and techniques of information security.
- Ability to analyze cyber threats and assess risks.
- Skills necessary to implement security and protection strategies.

- Recommended Textbooks:

- "Principles of Information Security" by Michael E. Whitman and Herbert J. Mattord.

- Prerequisites:

It is preferred that students have a basic understanding of operating system concepts and computer networks.

- Lab Topics:

None

Course Code (CNDC407)
Course Title: Mobile Computing
Credit Units - Lecture Hours - Practical Hours - Lab Hours (3-2-0-2)

- Course Definition:

The Mobile Computing course focuses on studying the applications and techniques of designing and developing mobile applications on smart devices such as smartphones and tablets.

- Course Topics

- Introduction to Mobile Computing; Elements of Mobile Computing
- Wireless Duplexing Techniques for Communications
- Multiple Access Techniques: Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA)
- GSM (Global System for Mobile Communications) (2G)
- UMTS (Universal Mobile Telecommunications Systems) (3G)
- Universal Subscriber Identity Module (USIM)
- Radio Network Subsystem (RNS)
- UMTS Radio Access Network (UTRAN)
- Cloud Computing: Evolution of Cloud Computing, Characteristics of Cloud Computing
- Cloud Computing Service Models: Software as a Service, Platform as a Service, Infrastructure as a Service.
- Cloud Computing Deployment Models: Public Cloud, Private Cloud, Hybrid Cloud
- Mobile Cloud Computing:
- Motivation for Mobile Cloud Computing
- Architecture of Mobile Cloud Computing: Service-Oriented Architecture, Agent-Client Architecture, Collaborative Architecture
- Mobile Internet Protocol: Overview, Features, and Mechanisms
- Mobile Ad Hoc Networks: Characteristics of MANETs, Applications of MANETs
- Long-Term Evolution (LTE) (4G)
- 5G Networks and Beyond

- Course Description:

The Mobile Computing course aims to introduce students to the fundamentals of designing and developing mobile applications, providing them with the necessary skills to build effective and innovative mobile applications.

- Course Outcomes

- Comprehensive understanding of the fundamentals of mobile application design and development.

- Ability to use various development tools and platforms to build mobile applications.
- Skills necessary for designing and implementing attractive and innovative user interfaces.

- Recommended Textbooks:
- "Mobile Computing" by Rajiv Yadhav and Emmanuel Frécon.

- Prerequisites:
There are no prerequisites, but it is preferable for students to have a basic understanding of computer programming and web development.

- - Lab. Topics

- Introduction to Mobile Computing: Overview of mobile computing concepts, mobile devices, operating systems, and mobile application development frameworks.
- Mobile Application Development Environment: Setting up the development environment for mobile application development, including installing necessary software and tools.
- Designing User Interfaces for Mobile Applications: Designing user-friendly and visually appealing user interfaces for mobile applications, considering constraints and best practices for mobile devices.
- Fundamentals of Mobile Application Development: Introduction to mobile application development frameworks such as Android or iOS, understanding the application lifecycle, and creating simple mobile applications.
- Mobile Networks and Communications: Exploring mobile networking technologies, understanding wireless communication protocols, and implementing network-based functionalities in mobile applications.
- Mobile Data Management: Dealing with data storage and retrieval in mobile applications, using local storage, databases, and cloud services to efficiently manage data.
- Testing and Deploying Mobile Applications: Testing mobile applications for functionality, performance, and usability, and deploying mobile applications to app stores or testing environments.

Course Code (CNDC408)
Network Protocols & Services
Credit Units - Lecture Hours - Practical Hours - Lab Hours (3-2-0-2)

- Course Definition

Network Protocols & Services course focuses on studying the protocols used for data exchange across networks and the various services provided over networks.

- Course Topics

- Networks, Protocols, and Services: Definition and Overview.
- OSI Seven-Layer Model.
- TCP/IP Four-Layer Model.
- Network Engineering Models: IBM SNA Model.
- Application Layer Protocols.
- Bootstrap Protocol (BOOTP).
- Dynamic Host Configuration Protocol (DHCP).
- Domain Name System (DNS) Protocol.
- File Transfer Protocol (FTP).
- Hypertext Transfer Protocol (HTTP).
- Network Time Protocol (NTP).
- Remote Monitoring (RMON) MIBs (RMON1 and RMON2).
- Simple Mail Transfer Protocol (SMTP).
- Network Management: Configuration Management, Fault Management, Performance Management.
- Presentation Layer Protocols.
- Lightweight Presentation Protocol (LPP).
- Session Layer Protocols.
- Remote Procedure Call (RPC).
- Transport Layer Protocols.
- Reliable Data Protocol (RDP).
- Transmission Control Protocol (TCP).
- User Datagram Protocol (UDP).
- Network Layer Protocols.
- Internet Protocol (IP) (IPv4).
- Open Shortest Path First (OSPF).
- Routing Information Protocol (RIP2).
- Data Link Layer Protocols.
- Address Resolution Protocol (ARP) and Inverse ARP.
- Internet Protocol Control Protocol (IPCP) and IPv6 Control Protocol (IPv6CP).
- Reverse Address Resolution Protocol (RARP).
- Serial Line Internet Protocol (SLIP).

- Course Description:

Network Protocols & Services course aims to introduce students to the concepts and protocols of different networks and provide them with the necessary knowledge to configure and manage computer networks effectively.

- Course Outcomes

- Deep understanding of various network protocols and their operations.
- Ability to configure and manage computer networks using appropriate protocols.
- Skills required to analyze and diagnose network problems and perform necessary maintenance.

- Recommended Textbooks:

- "Computer Networking: A Top-Down Approach" by James Kurose and Keith Ross.
- "TCP/IP Illustrated, Volume 1: The Protocols" by W. Richard Stevens.
- "Network Security Essentials" by William Stallings.

- Prerequisites:

It is preferable for students to have a basic understanding of networking concepts and operating systems.

- Lab. Topics

- Introduction to Network Protocols:
- Overview of network protocols, including the TCP/IP suite
- OSI model
- key protocols such as HTTP, DNS, and FTP.
- Network Protocol Analysis: Using network analysis tools such as Wireshark to capture and analyze network traffic,
- understanding protocol headers, and identifying network problems.

Course Code:UOA019

Course Title: Research Methodology
Credit Units - Lecture Hours - Practical Hours - Lab Hours: 2-2-0-0

- - **Course Definition**

Research Methodology course focuses on studying the methods and techniques used in planning and conducting scientific research in various fields.

- **Course Topics**

- Definition of Research Methodology
- Formulating Research Problems
- Identifying Research Objectives
- Literature Review
- Developing Research Hypotheses
- Research Design Preparation
- Data Collection
- Data Analysis
- Project Implementation
- Hypothesis Testing
- Generalization and Interpretation
- Data Analysis
- Report Preparation
- Presentation of Results

- **Course Description**

Research Methodology course aims to introduce students to the fundamentals of scientific research and equip them with the necessary skills to design, conduct, and analyze scientific research in a systematic and scientific manner.

- - **Course Outcomes**

- Deep understanding of the stages and methods of scientific research.
- Ability to design, conduct, and analyze scientific research.
- Skills required to write research reports and scientific articles.

- Recommended Textbooks:
- "Research Methodology: Methods and Techniques" by Krishna Kant.

- **Prerequisites:**
  There are no prerequisites.

Course Code: CNDC412
Course Title: Web Application Development I
Credit Units - Lecture Hours - Practical Hours - Lab Hours: 3-2-0-2

- - **Course Definition**

Web Application Development I course focuses on studying the design and development of web-based applications using programming languages such as HTML, CSS, and JavaScript.

- - **Course Topics**

- Introduction to Web Development: Overview of web development concepts, including client-side and server-side programming, ASP.NET configurations, and state management techniques.
- ASP.NET State Management Part I: Exploring state management techniques in ASP.NET such as view state and session state to preserve data across multiple requests.
- ASP.NET State Management Part II: Using cookies and caching mechanisms in ASP.NET to store and manage user data and optimize application performance.
- ASP.NET Web Controls Part I: Introduction to basic web controls in ASP.NET such as label, button, and text box to create interactive web applications.
- ASP.NET Web Controls Part II: Implementing advanced web controls including dropdown list, menu control, checkbox, radio button, and hyperlink button to enhance user interaction.
- ASP.NET Web Controls Part III: Utilizing additional web controls such as image control, calendar control, and tree view to create dynamic and attractive web interfaces.
- Client-Side vs. Server-Side Programming: Understanding the differences between client-side and server-side programming and their roles in web development.
- ASP.NET Expressions Part I: Using conditional expressions such as if-else and switch-case, and handling exceptions in ASP.NET applications.

- ASP.NET Expressions Part II: Implementing loop structures including for loop, foreach loop, and while loop to iterate data and perform repetitive tasks in ASP.NET.
- ASP.NET Data Collections Part I: Working with collection classes such as ArrayList and HashTable to store and manipulate related data sets in ASP.NET applications.
- ASP.NET Data Collections Part II: Using data structure Stack and Queue to manage data in a First-In-First-Out (FIFO) or Last-In-First-Out (LIFO) manner in ASP.NET.
- ASP.NET Data Collections Part III: Exploring generic data structure and data list to organize and efficiently access data elements in ASP.NET projects.
- Web Application Project: Applying the taught concepts to develop a practical web application project using ASP.NET technologies.

- Course Description:

The Web Application Development II course aims to teach students how to build and design professional and effective web applications. The focus is on applying theoretical concepts in practical application projects.

- Course Outcomes

- Deep understanding of the technologies and tools used in web application development.
- Ability to design, implement, and develop web applications professionally.
- Skills required to interact with and handle databases through web applications.

- Recommended Textbooks:

- "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst.
- "JavaScript and JQuery: Interactive Front-End Web Development" by Jon Duckett.

- Prerequisites:

Students are expected to have a basic understanding of programming languages such as HTML and CSS.

- Lab Topics:

- Introduction to Web Technologies: Overview of web development technologies including HTML, CSS, JavaScript, and understanding client-server architecture in web applications.
- HTML and CSS: Creating web pages using HTML tags, styling web pages using CSS, and understanding the structure and presentation of web content.
- Basics of JavaScript: Introduction to JavaScript programming language, understanding variables, data types, operations, control flow, and basic functions in JavaScript.
- Document Object Model (DOM): Dynamically manipulating HTML elements and styles using JavaScript, understanding DOM tree structure, and handling events.
- Web Forms and User Input: Creating web forms for user input and data validation, handling form submission, and processing user data on the server.
- Web Application Deployment: Deploying web applications to a web server, configuring server-side environments, and basic tasks for server management.
- Introduction to Server-Side Programming: Basics of server-side programming using a programming language such as PHP or Python, processing form data on the server, and creating dynamic web content.

Course Code: CCIT067
Course Title: Artificial Intelligence II
Credit Units - Lecture Hours - Practical Hours - Lab Hours: 3-2-0-2

- **Course Definition**

Artificial Intelligence II course is a continuation of the study in the field of artificial intelligence, focusing on advanced topics in machine learning, artificial intelligence, and research strategies.

- **Course Topics**

- Consultative Search: Heuristic functions.
- Hill Climbing Algorithm.
- Best-First Search Algorithm.
- Cost Functions.
- A* Algorithm.
- Properties of Consultative Functions.
- Game Playing: An Introduction.
- Minimax Algorithm.

- Alpha-Beta Pruning.
- Enhancements to Game Search.
- Expert Systems: Structure; Rule-Based Expert Systems.
- Rule-Based Production System Control Strategies: Backward Chaining and its Execution.
- Pure Forward Chaining and its Execution; Offensive and Defensive Control Strategies and their Execution.
- Uncertainty in Expert Systems: Representing Probabilities in Rules; Evidence Integration.
- Another Approach to Expert System Design: Decision Networks; "And-Or-Not" Networks.

- Course Description:
The Artificial Intelligence II course aims to deepen students' knowledge and understanding in the field of artificial intelligence, introducing them to advanced topics and applications in this field.

- Course Outcomes

- Deep understanding of advanced concepts and techniques in artificial intelligence.
- Ability to apply and implement models and techniques of artificial intelligence in practical projects.
- Skills necessary for analyzing and designing advanced intelligent systems.

- Recommended Textbooks:
- "Deep Learning" by Ian Goodfellow and Yoshua Bengio.
- "Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto.

- Prerequisites:
It is preferable for students to have a basic understanding of artificial intelligence and machine learning concepts.

- - Lab. Topics

- Reinforcement Learning: Introduction to reinforcement learning algorithms, including Q-learning, policy gradient, and deep reinforcement learning.

- Natural Language Processing: Building upon the foundations of natural language processing from Artificial Intelligence I, covering advanced topics such as sequence-to-sequence models, attention mechanisms, and language translation.
- Computer Vision: Introduction to computer vision techniques, including image classification, object detection, and image segmentation using Convolutional Neural Networks (CNNs).
- Neural Networks: Introduction to neural networks and deep learning, implementing feedforward networks and recurrent networks for classification tasks.
- Problem Solving and Search: Implementing search algorithms such as depth-first search, breadth-first search, and A* search for problem-solving.
- Knowledge Representation and Logic: Representing knowledge using propositional logic and first-order logic, implementing inference algorithms.
- Machine Learning: Introduction to machine learning algorithms, including supervised learning, unsupervised learning, and reinforcement learning.

Course Code: CNDC404
Course Title: Network Security
Credit Units - Lecture Hours - Practical Hours - Lab Hours: 2-2-0-0

- - Course Definition

Network Security focuses on studying policies and technologies used to protect networks and data from cyber-attacks and threats.

- Course Topics

- Principles I: Benefits of Switching in Networks, Drawbacks of Switching in Networks, Benefits of Routing in Networks, Drawbacks of Routing in Networks, Differences between Switching and Routing in Networks.
- Principles II: Why Do We Use Switching and Routing, Internal Structure of Switching, Internal Structure of Routing, Operation of Switching and Routing.
- Routing and Switching Strategies - Switching: Traffic Routing and Filtering.
- Routing and Switching Strategies - MAC Address-Based Routing.
- Routing: Path Finding, Routing Devices, Static Routes, Default Routes, Dynamic Routes.

- Routing Protocols I: Single vs. Multi, Internal vs. External.
- Routing Protocols II: Flat vs. Hierarchical, Link-State vs. Distance-Vector.
- Route Selection or Installation, Prefix Length, Administrative Distance Metric.
- Spanning Tree and Rapid Spanning Tree, Spanning Tree Structure, Why Loops Are Bad? Comparison Algorithm.
- Spanning Tree and Rapid Spanning Tree Messages, Issues with Spanning Tree, Transitioning from Switching to Switching: Special Case.
- VLAN Networks and Spanning Tree, Rapid Spanning Tree Protocol.
- VLAN Networks and Convergence: Broadcast Domains, What is a VLAN? Impact of Virtual Local Area Networks.
- Types of Virtual Local Area Networks, VLANs Between Switches.

- - **Course Description**:

The Network Switching and Routing course aims to introduce students to the fundamentals of packet switching and routing in networks and equip them with the necessary knowledge to design, implement, and manage networks effectively.

- - **Course Outcomes**:

- Understanding the principles and techniques of network switching and routing.
- Ability to analyze and evaluate different switching and routing strategies.
- Skills necessary to design and implement efficient network switching and routing solutions.

- What is Convergence? Convergence Protocol Criteria Pruning, Design Considerations for VLAN Networks.

- - **Course Outcomes**

- Deep understanding of network switching and routing concepts.
- Ability to analyze, design, and configure networks using appropriate switching and routing techniques.
- Skills necessary to analyze and troubleshoot network problems and perform required maintenance.

- **Recommended Textbooks:**
  - "Computer Networking: A Top-Down Approach" by Jim Kurose and Keith Ross.
  - "Routing TCP/IP, Volume 1" by Jeff Doyle.

- **Prerequisites:**
  It is preferable for students to have a basic understanding of networking concepts and protocols.

- **Lab. Topics**

- Introduction to Packet Tracer Software
- Switching in Packet Tracer
- Routing in Packet Tracer
- Network Address Translation (NAT) in Packet Tracer
- Quality of Service (QoS) in Packet Tracer
- Wide Area Networks (WANs) in Packet Tracer
- Dynamic Host Configuration Protocol (DHCP) in Packet Tracer

Course Code (CCIT068):
Course Title: Project
Credit Units - Lecture Hours - Practical Hours - Lab Hours (6--0-12)

- - **Course Definition**

The Graduation Project in Computer Network Systems is an opportunity for students to apply the concepts and skills they have acquired during their study period to a practical project in the field of networks.

- - **Course Topics**

:

- Needs Analysis and System Design for solving a specific problem or developing a specific system.
- Implementation and configuration of the network infrastructure.
- Testing and debugging errors.
- Documentation and presentation of results.

- - **Course Description**

The Graduation Project in Computer Network Systems aims to provide students with an opportunity to apply and expand their knowledge and skills in the field of networks through the design and implementation of a practical project.

- - **Course Outcomes**:

- Ability to analyze network needs and design an integrated network system.
- Skills necessary to implement, configure, test, and debug network infrastructure.
- Ability to deal with network problems and perform necessary maintenance.

- Recommended Textbooks:
- Varies depending on the specific problem to be solved or system to be developed.

- Prerequisites:
It is expected that students have a comprehensive understanding of networking concepts and techniques.

- - **Lab. Topics**

The project typically includes a practical component that allows students to apply the theories and techniques taught in previous courses. Supervision is provided directly by the student's advisor.

Course Code: CNDC405
Course Title: Web Application Development II
Credit Units - Lecture Hours - Practical Hours - Lab Hours: 3-2-0-2

- - **Course Definition**:

Web Application Development II is a continuation of the study in the field of web application development. This part focuses on advanced topics and techniques in designing and developing web applications.

- - **Course Topics:**

- Introduction
- Data Access in ASP.NET 1:
  - ADO.NET architecture
  - Advantages of ADO.Net
- Data Access in ASP.NET 2:
  - Disconnected data access architecture
  - ASP.NET connection strings
  - First ASP.NET database program
- ASP.NET Data Providers 1:
  - ASP.NET connection
  - ASP.NET SQL Server connection
  - ASP.NET OLEDB connection
  - ASP.NET ODBC connection
- ASP.NET Data Providers 2:
  - ASP.NET commands
  - ASP.NET ExecuteNonQuery
  - ASP.NET ExecuteScalar
  - ASP.NET ExecuteReader
- ASP.NET Data Providers 3:
  - ASP.NET data reader
  - ASP.NET data adapter
  - ASP.NET data adapter commands
- Authentication and Security:
  - Introduction to authentication and authorization
  - Working with JSON Web Tokens (JWT)
  - Best practices for securing web applications
- ASP.NET Dataset:
  - ASP.NET Dataset 1:
    - Using ASP.Net Dataset
    - Finding tables in Dataset
  - ASP.NET Dataset 2:
    - Number of rows in ASP.NET Dataset
    - Using dynamic ASP.NET Dataset
    - Defining columns in Dataset
- ASP.NET Database Programming:
  - ASP.NET Database Programming 1:
    - DBNull value in ASP.NET
    - Single quotation marks in ASP.NET
  - ASP.NET Database Programming 2:
    - ASP.NET Temporary Stored Procedures

• ASP.NET Parameterized Queries
• ASP.NET Database Programming 3:
• Recordset scope from database
• Image to ASP.NET database
- Application Project

- - **Course Description**:

The Web Application Development II course aims to expand students' knowledge and skills in designing and developing web applications through exploring advanced concepts and techniques.

- - **Course Outcomes**:

- Ability to apply popular frameworks in web application development.
- Proficiency in developing professional and advanced web applications using advanced techniques.
- Deep understanding of security concepts and techniques in web application development.

- **Recommended Textbooks:**
- "Flask Web Development: Developing Web Applications with Python" by Michael Grinberg

- Prerequisites:
Students are expected to have prior knowledge of web application development and programming languages used in this field.

- **Lab. Topics**

- Advanced Frontend Development: Exploring advanced frontend development techniques using HTML, CSS, and JavaScript frameworks like React or Angular.
- Server-Side Programming: Learning server-side programming languages like Node.js or PHP to build dynamic and interactive web applications.
- Integration with Databases: Integrating databases into web applications using technologies like MySQL, MongoDB, or PostgreSQL, and learning database design and queries.
- Authentication and Authorization: Implementing user authentication and authorization mechanisms in web applications, such as using OAuth, JWT, or session-based authentication.

- **Web Security: Understanding common web security vulnerabilities and best practices for securing web applications, including input validation, preventing cross-site scripting (XSS), and SQL injection prevention.**
- **Web Services and APIs: Exploring how to consume and create web services and APIs, including RESTful APIs and integrating with external services.**
- **Testing and Deployment of Applications: Learning various testing strategies for web applications, such as unit testing, integration testing, and end-to-end testing, and understanding the process of deploying web applications in production environments.**