B. TECH IN COMPUTER SCIENCE AND ENGINEERING

PROGRAM EDUCATIONAL OBJECTIVES

- Prepare and train students in theoretical foundations to work with cutting edge computing technologies and design solutions to complex engineering problems, making them ready to work in industrial environment.
- Develop all round skills such as team building, inter-personal skills, and leadership qualities in order to effectively communicate with engineering community and with society at large.
- Promote research culture through internships, research assistantships, research-oriented projects, sponsored and collaborative research and enable them to pursue higher studies in computer science and related fields.
- To inculcate social concern meeting the requirements of prospective employers and to develop an ability to innovate efficient computing solutions for a better society.
- Create professionally superior and ethically strong globally competent employees and entrepreneurs.

PROGRAM OUTCOMES

- Apply mathematical and theoretical principles in the modelling and design of high-quality computer-based systems using state-of-the-art computer technology.
- Conduct in-depth study of research literature in the area of Computer Science, analyse problems in order to arrive at substantiated conclusions using first principles of mathematics, and allied sciences.
- Design, implement and evaluate Computer Systems, programs and processes that meet partial/ complete specifications with concern for society, environment and culture.
- Design and conduct experiments, collect data, analyze and interpret the results to investigate complex engineering problems in the field of Computer Science.
- Apply state-of-the-art techniques and modern computer-based tools in prediction, comparison and modelling of complex engineering activities.
- Have a sound understanding of professional, legal, security and social issues and responsibilities in engineering activities involving Computer Science.
- Understand societal and environmental concerns and demonstrate responsibility in sustainable development of computer-based solutions.
- Be aware of ethical and professional responsibilities in engineering situations; make informed judgments regarding intellectual property and rights in relation to computer-based solutions in global, economic, environmental and societal contexts.
- Able to function effectively in teams to establish goals, plan tasks, meet deadlines, manage risk and produce high-quality technical solutions.
- Contribute and communicate effectively with the society, be able to write effective reports and design documents by adhering to appropriate standards, make effective presentations, give and receive clear instructions.
- Apply skills in clear communication, responsible teamwork and time management by, for example, managing a team or project and communicating with external stakeholders.
- Recognize the need for and demonstrate an ability to engage in continuing professional development in its broadest sense.

B. TECH IN COMPUTER SCIENCE AND ENGINEERING
I SEMESTER (2024-28 BATCH)

Sl. No.	Course Code	Course Title	Hours per week				Credits	Tools / Languages	Course Type
			L	T	P	S	C		
1	UE24CS151A	Python for Computational Problem Solving	4	0	2	5	5	Python interpreter 3.8 and above. IDLE and any IDE like Jupyter, coding platforms like HackerRank	FC- Lab Integrated

II SEMESTER (2024-28 BATCH)

Sl. No.	Course Code	Course Title	Hours per week				Credits	Tools / Languages	Course Type
			L	T	P	S	C		
1	UE24CS151B	Problem Solving with C	4	0	2	5	5	C Programming Language, GCC Compiler, GDB DebuggerLab , coding platforms like HackerRank	FC- Lab Integrated

III SEMESTER (2023-27 BATCH)

Sl. No.	Course Code	Course Title	Hours per week				Credits	Tools / Languages	Course Type
			L	T	P	S	C		
1	UE23CS251A	Digital Design and Computer Organization	4	0	2	5	5	Icarus Verilog Simulator, GTK Wave waveform viewer. (Open–Source Tool)	CC- Lab Integrated
2	UE23CS252A	Data Structures and its Applications	4	0	2	5	5	C Programming Language, Coding Platforms like HackerRank	CC-Lab Integrated
3	UE23MA242A	Mathematics for Computer Science and Engineering	4	0	0	4	4	Jupyter Notebook**,** Python, Pandas, Matplotlib, Scipy, Seaborn, BeautifulSoup, Numpy, Scikit learn.	CC- Independent
4	UE23CS241A	Web Technologies	4	0	0	4	4	HTML, CSS, JavaScript, MERN Technologies.	CC- Independent
5	UE23CS242A	Automata Formal Languages and Logic	4	0	0	4	4	JFLAP	CC- Independent
6	UE23EC221A	CIE L1	2	0	0	2	2		CC- Independent
7	UE24MA221A*	Bridge Course Mathematics –I (Applicable for the Lateral Entry Students)	2	0	0	2	0		FC- Independent
Total			**22/24**	**0**	**4**	**24/26**	**24**		
* - Audit Course									

IV SEMESTER (2023-27 BATCH)

Sl. No.	Course Code	Course Title	Hours per week				Credits	Tools / Languages	Course Type
			L	T	P	S	C		
1	UE23CS251B	Microprocessor and Computer Architecture[%]	4	0	2	5	5	ARM Simulator, Arduino Microcontroller kit, ParaCache simulator.	CC-Lab Integrated
2	UE23CS252B	Computer Networks	4	0	2	5	5	Wireshark, Python	CC-Lab Integrated
3	UE23CS241B	Design and Analysis of Algorithms	4	0	0	4	4	C Programming Language, GCC Compiler.	CC-Independent
4	UE23CS242B	Operating Systems[@]	4	0	0	4	4	C, Linux/Unix OS for system call implementation.	CC-Independent
5	UE23MA241B	Linear Algebra and its Applications	4	0	0	4	4	Python IDE	CC-Independent
6	UE23EC221B	CIE L2	2	0	0	2	2		CC-Independent
7	UE24MA221B *	Bridge Course Mathematics –II (Applicable to Lateral Entry Students)	2	0	0	2	0		FC-Independent
Total			**22/24**	**0**	**4**	**24/ 26**	**24**		

Note: Desirable Knowledge - [%]UE23CS251A, [@]UE23CS252A.
* - Audit Course

V SEMESTER (2022-26 BATCH)

Sl. No.	Course Code	Course Title	Hours per week				Credits	Tools / Languages	Course Type
			L	T	P	S	C		
1	UE22CS351A	Database Management System	4	0	2	5	5	MySQL Workbench, Python, ERwin, Any other tool for ER modeling	CC-Lab Integrated
2	UE22CS352A	Machine Learning*	4	0	2	5	5	NetworkX for statistical features of graphs, Tensorflow, Keras, and Scikit Learn for traditional graph ML, and Pytorch Geometric for Graph Neural Networks.	CC-Lab Integrated
3	UE22CS341A	Software Engineering	4	0	0	4	4	GitHub, MS Project/GantPro/Jira, Jenkins.	CC-Independent
4	UE22CS342AAX	Elective I	4	0	0	4	4		EC
5	UE22CS343ABX	Elective II	4	0	0	4	4		EC
6	UE22CS320A	Capstone Project – Phase	0	0	8	8	2		

		1							
Total			**20**	**0**	**12**	**30**	**24**		
Elective – I									
6	UE22CS342AA1	Advanced Algorithms[%]	4	0	0	4	4	C or C++ Programming Language	EC-Independent
7	UE22CS342AA2	Data Analytics[&]	4	0	0	4	4	R andPython.	EC-Independent
8	UE22CS342AA3	Internet of Things[$]	4	0	0	4	4	Python, Embedded-C, Cloud Platforms, Single Board Computerss	EC-Independent
9	UE22CS342AA4	Applied Cryptography	4	0	0	4	4	SEED labs, Python Programming Language	EC-Independent
10	UE22CS342AA5	Augmented and Virtual Reality [!!!]	4	0	0	4	4	C/ C++/ JAVA/ Python using OpenGL.	EC-Independent
11	UE22CS342AA6	Human Computer Interaction	4	0	0	4	4	Figma, Adobe Creative Cloud.	EC-Independent
Elective – II									
12	UE22CS343AB1	Enterprise Business Systems for Engineers	4	0	0	4	4	Oracle, Salesforce, Open ERP.	EC-Independent
13	UE22CS343AB2	Big Data [$]	4	0	0	4	4	Hadoop, HDFS, Spark, Streaming Spark, HIVE, HBase, Mllib.	EC-Independent
14	UE22CS343AB3	Graph Theory, and its Applications [!]	4	0	0	4	4	Python,Neo4j, NetworkX	EC-Independent
15	UE22CS343AB4	Bio-inspired Computing [%]	4	0	0	4	4	Matlab, Inspyred pythonic tool	EC-Independent
16	UE22CS343AB5	Advanced Computer Networks[%%]	4	0	0	4	4	GNS3, Cisco Packet Tracer.	EC-Independent
17	UE22CS343AB6	Computer Network Security[%%]	4	0	0	4	4	SEED Ubuntu VM, Wireshark, Snort, Netwox, Scapy.	EC-Independent
18	UE22CS343AB7	Robotics with ROS	4	0	0	4	4	C++, Python, Java, ROS, Rviz, Gazebo and Webots.	EC-Independent

Note: Desirable Knowledge

Core : [*]- UE22CS241A, UE22MA241B, UE22CS241B.

Elective I : [%]- UE22CS241B, [&]- UE22CS241A, [$] - UE22CS151A, UE22CS151B, UE22EC141A, UE22CS252B, [!!!]- UE22CS252A.

Elective II :[$]-UE22CS252A, UE22CS241B, [!]- UE22CS151B, UE22CS252A, [%]- UE22CS241B, [%%]- UE22CS252B.

ELECTIVES TO BE OPTED FOR SPECIALIZATION			
Sl. No.	**SPECIALIZATION**	**ELECTIVE – I**	**ELECTIVE – II**
A	System and Core Computing(SCC)	UE22CS342AA1, UE22CS342AA5.	UE22CS343AB1, UE22CS343AB2, UE22CS343AB3.
B	Machine Intelligence and Data Science(MIDS)	UE22CS342AA2, UE22CS342AA3, UE22CS342AA5, UE22CS342AA6	UE22CS343AB2, UE22CS343AB3, UE22CS343AB4
C	Cyber Security & Connected Systems (CSCS)	UE22CS342AA3, UE22CS342AA4, UE22CS342AA5.	UE22CS343AB5, UE22CS343AB6. UE22CS343AB7

VI SEMESTER (2022-26 BATCH)

Sl. No	**Course Code**	**Course Title**	**Hours per week**				**Credits C**	**Tools / Languages**	**Course Type**
			L	**T**	**P**	**S**			
1	UE22CS351B	Cloud Computing[@@]	4	0	2	5	5	Amazon AWS (or equivalent), AWS Skill Builder, AWS Educate, Qwiklabs, Docker, Kubernetes, Jenkins, Zookeeper, Github, NoSQL database, Flask, Python, Go Lang.	CC
2	UE22CS352B	Object Oriented Analysis and Design	4	0	2	5	5	Star UML, Object Oriented Programming Language (Java/C++)	CC
3	UE22CS341B	Compiler Design[!]	4	0	0	4	4	Lex/flexandYACC/ Bison.	CC
4	UE22CS342BAX	Elective III	4	0	0	4	4		EC
5	UE22CS343BBX	Elective IV	4	0	0	4	4		EC
6	UE22CS320B	Capstone Project Phase-2	0	0	24	24	2		PW
Total			**20**	**0**	**26**	**46**	**24**		
Elective – III									
7	UE22CS342BA1	Supply Chain Management for Engineers	4	0	0	4	4	SCM applications and tools, ML tools, Case studies, Web resources.	EC-Independent
8	UE22CS342BA2	Algorithms for Information Retrieval	4	0	0	4	4	Scikit, Tensorflow, Solr, Lucene Search Engines/ Python Programming Languages	EC-Independent
9	UE22CS342BA3	Image Processing and Computer Vision **	4	0	0	4	4	MatLab, Python Programming Languages	EC-Independent

10	UE22CS342BA4	Natural Language Processing ##	4	0	0	4	4	Tensorflow, Scikit Learn, Python 3.x. CoreNLP, Natural Language Toolkit (NLTK), TextBlob,Gensim, SpaCy,PyTorch-NLP , OpenNLP, Hugging Face, OpenAI API.	EC-Independent
11	UE22CS342BA5	BlockChain !	4	0	0	4	4	Solidity, Remix, Ganache, Metamask.	EC-Independent
12	UE22CS342BA6	Digital Forensics and Incident Response	4	0	0	4	4	Open source tools on Forensics.	EC-Independent
13	UE22CS342BA7	Digital Twin and eXtended Reality !	4	0	0	4	4	C/ C++/ JAVA/ Python using OpenGL.	EC-Independent
14	UE22CS342BA8	Topics in Wireless Networks and 5G	4	0	0	4	4	Wireshark, Claynet, Cisco Packet Tracker.	EC-Independent
15	UE22CS342BA9	Generative AI and its Applications	4	0	0	4	4	Python ,HuggingFace, LM Studio, Kaggle	EC-Independent
16	UE22CS342BA10	Cloud Security	4	0	0	4	4	AWS Security Services	EC-Independent
Elective – IV									
17	UE22CS343BB1	Heterogeneous Parallelism !!!	4	0	0	4	4	pthread, OpenMP CUDA, openCL.	EC-Independent
18	UE22CS343BB2	Topics in Deep Learning ##	4	0	0	4	4	Pytorch.	EC-Independent
19	UE22CS343BB3	Database Technologies***	4	0	0	4	4	MySQL, postgres, Oracle, Apache Spark, Apache Kafka, Amazon Kinesis.	EC-Independent
20	UE22CS343BB4	Machine Learning on Graphs %%%	4	0	0	4	4	NetworkX for statistical features of graphs,Tensor flow Keras and Scikit Learn for traditional graph ML, and Pytorch Geometric for Graph Neural Networks.	EC-Independent
21	UE22CS343BB6	Information Security	4	0	0	4	4	SSEED Labs VM, Scapy, Burp Suite, Metasploit, Nmap, etc.	EC-Independent
22	UE22CS343BB7	Mobile and Autonomous Robotics	4	0	0	4	4	C, C++, Python, ROS	EC-Independent
23	UE22CS343BB8	Security for Internet of Things.	4	0	0	4	4	Wireshark, Yersinia, VoIP Hopper, Bettercap,aircrack-ng	EC-Independent
24	UE22CS343BB9	Applied ML in IoT with TinyML	4	0	0	4	4	Arduino Nano 33 BLE sense Arduino IDE Google Colab	EC-Independent

Note: Desirable Knowledge
Core: [!]UE22CS252A, UE22CS243A [@@]UE22CS241B, UE22CS252B.

Elective III:[#] - UE22CS151A, UE22CS151B, UE22CS252A, UE22CS252B, UE22CS241B, [??] - UE22CS241B, UE22MA241B, UE22CS352A [**]- UE22CS241B, [##] - UE22CS352A, [!] - UE22CS252A.

Elective IV: [!!!]-UE22CS151B, UE22CS251B, [##]-UE22CS352A, *** - UE22CS351A, [%%%] - UE22CS343AB3, UE22CS352A, [%%] - UE22CS252B.

ELECTIVES TO BE OPTED FOR SPECIALIZATION

Sl. No.	SPECIALIZATION	ELECTIVE – III	ELECTIVE – IV
A	System and Core Computing (SCC)	UE22CS342BA1, UE22CS342BA2, UE22CS342BA7	UE22CS343BB1, UE22CS343BB3.
B	Machine Intelligence and Data Science (MIDS)	UE22CS342BA2, UE22CS342BA3, UE22CS342BA4, UE22CS342BA7, UE22CS342BA9	UE22CS343BB2, UE22CS343BB4, UE22CS343BB7 UE22CS343BB9
C	Cyber Security & Connected Systems (CSCS)	UE22CS342BA5, UE22CS342BA6, UE22CS342BA7, UE22CS342BA8, UE22CS342BA10.	UE22CS343BB6, UE22CS343BB7, UE22CS343BB8.

VII SEMESTER (2021-25 BATCH)

SI No.	Course Code	Course Title	Hours / week				Credits	Course Type
			L	T	P	S	C	
1	UE21CS461A	Capstone Phase-2	0	0	24	24	6	PW
2	U21CS421AX	Special Topic/ Directed Independent Study - Swayam/MOOC	2/4/4			2	4	ST
Total			**2/4/4**	**2/4/4**	**24**	**26**	**10**	

VIII SEMESTER (2021-25 BATCH)

Sl. No.	Course Code	Course Title	Hours / week				Credits	Course Type
			L	**T**	**P**	**S**	**C**	
1	UE21CS421B	Capstone Phase-3	0	0	8	8	2	PW
2	UE21CS461BX	Internship	2	0	2	2	6	ST
3	UE21CS421BX	Special Topic/ Directed Independent Study – Swayam/MOOC	2	0	0	2	2	
Total			**2**	**0**	**10**	**10**	**10**	

UE24CS151A: Python for Computational Problem Solving (4-0-2-5-5)

Python is an easy to learn, general-purpose, powerful programming language. It has efficient high- level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

Course Objectives:

- Learn the syntax and semantics of Python programming language.
- Illustrate the process of structuring the data using lists, tuples, sets and dictionaries.
- Demonstrate the use of built-in functions to navigate the file system.
- Learn various paradigms of programming and implement the Object-Oriented Programming concepts in Python.

Course Outcomes:

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

- Program effectively using the Python language.
- Identify the methods to create and manipulate lists, tuples and dictionaries.
- Discover commonly used operations involving file system.
- Think using different paradigms of programming and interpret the concepts of Object- Oriented Programming as used in Python.

Course Contents:

Unit 1: Introduction

Computation Problem Solving-Limits of Computational Problem Solving - Computer Algorithm – Computer Hardware. Digital Computer - Operating System- Limits of IC technology - Computer Software - Syntax, semantics and program translation, Introduction to Python Programming Language, IDLE Python Development Environment, Output function - variables, types and id, input function, operators and expressions, Control structures.

14 Hours

Unit 2: Collections &Function

Lists, Tuples, Dictionaries, Sets, Strings and text file manipulation: reading and writing files. Functions: Definition, call Positional and keyword parameter, Default parameters, Variable number of arguments.

14 Hours

Unit 3: Functions, GUI, Modules, Testing and Debugging

Recursion, Call-backs, Closure, Decorators, generators. Graphical User Interface with Tkinter package- Different geometric methods – Tk, mainloop, Creating simple GUI - buttons, canvas, check button, labels, entry fields, dialogs Widgets - sizes, fonts, colours layouts, nested frames, Modules - import mechanisms. Testing- Pytest , Function testing with Doctest, pdb debugger commands.

14 Hours

Unit 4: Functional & Object Oriented Programming

Lambda function, Map, filter, and reduce, max, min, Zip, list comprehension.
Classes and objects - inheritance, polymorphism, iterators, Error handling & Exceptions - try, except and raise, exception propagation.

14 Hours

Tools / Languages: Python interpreter 3.8 and above. IDLE and any IDE like Jupyter.

Text Book(s):

1: Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", 2 nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at http://greenteapress.com/thinkpython2/thinkpython2.pdf (Chapters 13, 15, 16, 17, 18) (Download pdf/html files from the above link)
2: Al Sweigart,"Automate the Boring Stuff with Python",1 st Edition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/)
3: Reema Thareja, Python Programming using Problem Solving approach, Second Edition.

Reference Book(s):

1: “Introduction to Computer Science Using Python: A Computational Problem- Focus”, Charles Dierbach, Wiley India Edition,John Wiley, 2015.
2: “Learn python Programming”, Fabrizio Romano, 2nd Edition, Packet Publishing,2018.
3: “Fundamentals of Python: First Programs”, Kenneth A. Lambert, Cengage, 2019.
4: “Introduction to Computation and Programming Using Python: With Application to Understanding Data”,John V. Guttag, MIT Press, MIT with Library of Congress Cataloguing- in-PublicationData, 2016.

Lab / Hands-on: **14 Hours**

1. Programs on Input Output Functions, Operators and Expressions and Usage of Libraries
2. Programs on Control Structures.
3. Programs on Collections (Lists, Tuples, Sets , Dictionaries , Strings).
4. Programs on Files and File manipulations.
5. Programs on Functions.
6. Programs on Functional Programming.
7. Programs on Object Oriented Programming.
8. Programs on Pytest, pdb and GUI.

UE24CS151B: Problem Solving with C (4-0-2-5-5)

In Problem Solving with C Course, students will be learning to solve common types of computational problems, by analysing the given problem statement and developing an algorithm to solve the given problem. C language constructs are used to develop well-structured programs.

Course objectives:

- Acquire knowledge on how to solve relevant and logical problems using computing machine
- Map algorithmic solutions to relevant features of C programming language constructs
- Gain knowledge about C constructs and its associated ecosystem
- Appreciate and gain knowledge about the issues with C Standards and its respective behaviours.

Course outcomes:

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

- Understand and Apply algorithmic solutions to counting problems using appropriate C Constructs
- Understand, Analyze and Apply sorting and Searching techniques
- Understand, Analyze and Apply text processing and string manipulation methods using Arrays, Pointers and functions
- Understand user defined type creation and implement the same using C structures, unions and other ways by reading and storing the data in secondary systems which are portable.

Course Contents:

Unit1: Problem Solving Fundamentals

Introduction to Programming, Salient Features of 'C', Program Structure, Variables, Data Types & range of values ,Qualifiers, Operators and Expressions, Control Structures, Input/Output Functions, Language Specifications -Behaviors, Single character input and output, Coding standards and guidelines.

14 Hours

Unit2: Counting, Sorting and Searching

Arrays–1D and 2D, Pointers, Pointer to an array, Array of pointers, Functions, Call back, Storage classes, Recursion, Searching, Sorting.

14 Hours

Unit3: Text Processing and User-Defined Types

Strings, String Manipulation Functions & Error handling, Command line arguments, Dynamic Memory Management functions & Error handling, Structures, #pragma, Array of Structures, Pointer to structures, Passing Structure and Array of structure to a function, Bit fields, Unions, Enums, Lists, Statck, Queue, Priority Queue.

14 Hours

Unit4: File Handling and Portable Programming

File IO using redirection, File Handling functions of C, Searching, Sorting, Header files, Comparison of relevant User defined and Built-in functions, Variable Length Arguments, Environment variables, Preprocessor Directives, Conditional Compilation.

14 Hours

Tools/ Languages : C Programming Language, GCC Compiler, GDB Debugger, Coding platforms like HackerRank

Laboratory / Hands-on : **14 Hours**

1. Programs on IO, Operators and Control structures.
2. Programs on Arrays and Pointers, Functions using arrays and pointers
3. Programs on Strings, Structures and Dynamic Memory Management functions
4. Programs on Structures and Array of structures
5. Programs on Inclusion of files using redirection operator.
6. Programs using File handling functions in C - Sorting and Searching using Array of pointers to structures.
7. Implementation of Ordered List
8. Implementation of Priority based scheduling.
9. Programs on unions, enums and Preprocessor directives.

Text Book(s) :

1: "Computer Fundamentals and Programming in C", Reema Thareja, 2nd edition, Oxford University Press, 2016.

2.“The C Programming Language”, Brian Kernighan and Dennis Ritchie, Prentice Hall PTR, 2nd Edition, 1988.

Reference Book(s) :

1: “How To Solve It By Computer”, R G Dromey, Pearson, 2011.
2: “C Programming: A Modern Approach”, 2nd Edition by K.N. King. W. W. Norton & Company.
3: “Learn C the Hard Way”: Zed Shaw’s Hard Way Series, 1st Edition.

UE23CS251A : Digital Design and Computer Organization(4-0-2-5-5)

This course focuses on the structure, design and operation of a computer system at different levels of abstraction. The digital design part of the course describes low level digital logic building blocks while the computer organization part explains the structure and operation of microprocessors.

Course Objectives:

- Fundamental (combinational and sequential) building blocks of digital logic circuits.
- Design of more complex logic circuits such as adders, multipliers and register files.
- Design of Finite State Machines based on problem specification.
- Construction, using above logic circuits, of a microprocessor and its functioning at the clock cycle level.

Course Outcomes:

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

- Perform analysis of given synchronous digital logic circuit.
- Design and implement small to medium scale data path logic circuits from given specification.
- Design and implement control logic using Finite State Machines.
- Understand hardware level microprocessor operation, providing a foundation for the higher layers and utilize the concepts and techniques learnt to implement complex digital systems.

Course Content:

Unit 1: Gate-Level Minimization and Combinational logic-1

Introduction, The map method Four variable K-map, Product of Sums simplification, Don't Care conditions, NAND and NOR implementation, Combinational circuits, Analysis procedure Design Procedure, Combinational logic-1: Binary Combinational logic :Adder- Subtractor, Decimal Adder, Binary multiplier, Magnitude comparator Decoders Encoders, Multiplexers.

14 Hours

Unit 2: Synchronous Sequential Logic-I

Synchronous Sequential Logic: Introduction, Sequential circuits, Storage elements: Latches, Flip flops, Analysis of clocked sequential circuits, State reduction and assignment, Design procedure Registers and counters: Registers, Shift register, Ripple counters, Synchronous counters Other counters.

14 Hours

Unit 3: Basic structure of computers, Standard I/O interface, Interrupts

Computer Types, Functional Units: Input Unit, Memory Unit, ALU, Output Unit, Control Unit, Basic operational concepts ,Number representation and arithmetic Operations, Character representation, Memory locations and addresses, Memory Operations ,Instruction and instruction sequencing ,Addressing modes, Assembly Languages , I/O Operations: Accessing I/O Devices, Interrupts Standard I/O Interfaces.

14 Hours

Unit 4: Arithmetic Processing Unit and Control Unit Design

Arithmetic: Multiplication of Positive numbers, Signed operand Multiplication, Fast multiplication, Integer division, floating point numbers operation and architecture ,Some fundamental concepts, Execution of a complete instruction, Multiple Bus Organization, Hardwired control ,Single-cycle, Multi-cycle processor data path and control.

14 Hours

Tools / Languages: Icarus Verilog Simulator, GTK Wave waveform viewer. (Open–Source Tool)

Text Book(s):

1: "Digital Design", M Morris Mano, Michael D Ciletti, Pearson, 5th Edition, 2012.
2: "Computer Organization", Carl Hamacher, Zvonko Vranesic, Safwat Zaky, McGraw Hill, 5th Edition,2002.

Reference Book(s):

1: Digital Design & Computer Architecture, David Money Harris, Sarah L. Harris, 2nd Edition, Elsevier, 2013.
2: Computer Organization and Design, David A. Patterson, John L. Hennessey 5th Edition, Elsevier.

Lab/Hands-on: **14 Hours**

1: Verilog Basics- Basic gates, Adder/ Subtractor (One bit & n-bit).
2: Design of an ALU.
3: Design of a Register File.
4: Design of a Data path (Integration of ALU and Register File).
5: Design of a Program Counter.
6: Design of Control Logic.

UE23CS252A: Data Structures and its Applications (4-0-2-5-5)

This course introduces abstract concepts, shows how the concepts are useful for problem solving and then shows how the abstractions can be made concrete by using a programming language. Equal emphasis is placed on both the abstract and the concrete versions of a concept so that the student learns about the concept itself, its implementation and its application.

Course Objectives:

- Basic approaches and mindsets for analyzing and designing data structures and construct essential skills of data structures to store and retrieve data quickly and usefully (efficiently).
- Usage of different data structures that support different set of operations which are suitable for different type of tasks.
- Implement how to insert, delete, search and modify data in any given data structures- Stack, Queue, List, Tree, heap, Graphs.
- Implement a given application using the available data structure.

Course Outcomes:

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

- Choose relevant data structures for any given application
- Apply the required to implement any data structure.
- Appropriate data structure in competitive programming.
- Design and develop efficient software systems with good knowledge of data structures.

Course Content:

Unit 1: Linked List and Stacks

Review of C , Static and Dynamic Memory Allocation. Linked List: Doubly Linked List, Circular Linked List – Single and Double, Multilist: Introduction to sparse matrix (structure). Skip list Case study: Dictionary implementation using skip list Stacks: Basic structure of a Stack, Implementation of a Stack using Arrays & Linked list. Applications of Stack: Function execution, Nested functions, Recursion: Tower of Hanoi. Conversion & Evaluation of an expression: Infix to postfix, Infix to prefix, Evaluation of an Expression, Matching of Parenthesis.

14 Hours

Unit 2: Queues and Trees

Queues & Dequeue: Basic Structure of a Simple Queue, Circular Queue, Priority Queue, Dequeue and its implementation using Arrays and Linked List. Applications of Queue: Case Study – Josephus problem, CPU scheduling- Implementation using queue (simple /circular). General: N-ary trees, Binary Trees, Binary Search Trees (BST) and Forest: definition, properties, conversion of an N-ary tree and a Forest to a binary tree. Traversal of trees: Preorder, Inorder and Postorder.

14 Hours

Unit 3: Application of Trees and Introduction to Graphs

Implementation of BST using arrays and dynamic allocation: Insertion and deletion operations, Implementation of binary expression tree., Threaded binary search tree and its implementation. Heap: Implementation using arrays. Implementation of Priority Queue using heap - min and max heap. Applications of Trees and Heaps: Implementation of a dictionary / decision tree (Words with their meanings). Balanced Trees: definition, AVL Trees, Rotation, Splay Tree, Graphs: Introduction, Properties, Representation of graphs: Adjacency matrix, Adjacency list. Implementation of graphs using adjacency matrix and lists. Graph traversal methods: Depth first search, Breadth first search techniques. Application: Graph representation: Representation of computer network topology.

14 Hours

Unit 4: Applications of Graphs , B-Trees, Suffix Tree and Hashing

Application of BFS and DFS: Connectivity of graph, finding path in a network. Suffix Trees: Definition, Introduction of Trie Trees, Suffix trees. Implementations of TRIE trees, insert, delete and search operations. Hashing: Simple mapping / Hashing: hash function, hash table, Collision Handling: Separate Chaining & Open Addressing, Double Hashing, and Rehashing. Applications: URLs decoding, Word prediction using TRIE trees / Suffix Trees.

14 Hours

Tool/ Languages: C Programming Language

Lab / Hands-on: **14 Hours**

Implementation of:

1: Linked List and advanced operations.
2: Stack and applications based on it.
3: Queue and applications based on it.
4: Binary Tree, Binary Search Tree and applications based on it.
5: Graph Data structure and applications based on it.
6: Hashing Techniques.

Text Book(s):

1: "Data Structures using C / C++", Langsum Yedidyah, Moshe J Augenstein, Aaron M Tenenbaum Pearson Education Inc, 2nd edition, 2015.

Reference Book(s):

1: "Data Structures and Program Design in C", Robert Kruse, Bruce Leung, C.L Tondo, Shashi Mogalla, Pearson, 2nd Edition, 2019.

UE23MA242A: Mathematics for Computer Science and Engineering (4-0-0-4- 4)

This course covers both descriptive statistics to understand the data and inferential statistics which seeks to infer something about a population based on a statistical sample and build a simple linear regression model. This course will also cover engineering optimization techniques like unconstrained, constrained, and discrete optimization techniques.

Course Objectives:

- Provide insights about the basic roles of a Data Scientist. Understanding of Random Variables, and their Distributions.
- ProvidestudentswithknowledgeofConfidenceIntervalsandtheirimportance.Makeinferencesaboutthe population parameters using sample data and test it to draw meaningful conclusions.
- Provideanunderstandingontheimportanceandtechniquesofpredictingarelationship between the two sets of data and determine the goodness of fit model.
- Provideanunderstandingoftheimportanceandtechniquesof EngineeringOptimization.

Course Outcomes:

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

- UsePythonandothertoolstoextract,clean,andanalyzedatafromseveraldatasources(files,web)analyzean extremely large dataset, and perform exploratory data analysis to extract meaningful insights.
- Analyzeareal-worldproblemandsolvethesamewiththeknowledgegainedfromvariousdistributionstudies.
- Develop and test hypothesis about the population parameters to draw meaningful conclusions and fita regression model to data and use it for prediction.
- Analyze a real-world problem and solve the same with the knowledge gained from various engineering Optimization Techniques.

Course Content

Unit1: ApplicationsofProbabilityDistributionsandPrinciplesofPointEstimation

Introduction, Motivating Examples, and Scope. Statistics: Introduction, Types of Statistics, Types of Data, Types of Experiments – Controlled and Observational Study, Sampling: Sampling Methods, Sampling Errors, Case Study. Chebyshev's inequality, Normal Probability Plots, Introduction to Generation of Random Variants and mention the types, Acceptance-Rejection Method, Sampling Distribution, The Central Limit Theorem and Applications, Principles of Point Estimation - Mean Squared Error for Bernoulli, Binomial, Poisson, Normal, Maximum Likelihood Estimate for Bernoulli, Binomial, Poisson, Normal and Case Study. Introduction to Multivariate Normal Distribution, MAP Distribution.

Self-Learning: Generation of Random Variates -Inverse Transform Method.

Applications:

1. Poisson Distribution, calculation of the number of calls received in a specified time duration in call centers.
2. Variance, Standard Deviation, identifying customer satisfaction in online shopping.
3. Central Limit theorem, Load Balancing in distributed systems, and internet traffic prediction.
4. Sampling Mean, Estimating database query response times.

16 Hours

Unit2: Confidence Intervals and Hypothesis Testing

Confidence Intervals: Interval Estimates for Mean of Large and Small Samples, Student's t Distribution, Interval Estimates for Proportion of Large and Small Samples, Confidence Intervals for the Difference between Two Means, Interval Estimates for Paired Data. Factors affecting Margin of Error, Hypothesis Testing for Population Mean and Population Proportion of Large and Small Samples, Drawing Conclusions from the results of Hypothesis tests, and Case Study.

Self-Learning: Confidence Interval for the difference between two Proportions.

Applications:

1. t-distribution , Confidence Interval, Student's performance analysis based on hours of study.
2. z-test, application form processing in the banking system.
3. Hypothesis Testing, Randomly Trained Student's placement into tier-I and tier-II Companies.

12 Hours

Unit3:Distribution Free Tests and Multiple Linear Regression

Distribution Free Tests, Chi-Squared Test, Fixed Level Testing, Type I and Type II Errors, Power of a Test, Factors Affecting Power of a Test. Simple Linear Regression: Introduction, Correlation, Least Square Lines, Predictions using Regression Models

- Uncertainties in Regression Coefficients, Checking Assumptions and Transforming Data, Introduction to the Multiple Regression Model, Case Study.

Self-Learning:F test for equality of Variance.

Applications:

1. Linear regression, Stock Market prediction using Chi-Square Test, Analyzing the association between vaccination and recovery of the patients considering COVID data.
2. Chi-Square Test and Test of Independence, Analyzing the relationship between gender and preference for a product purchase.
3. Identifying Type 1 and Type 2 Errors in Spam Mail Classification.

14 Hours

Unit4:Engineering Optimization

Introduction to Optimization-Based Design, Modelling Concepts, Unconstrained Optimization, Discrete Variable Optimization, Genetic and Evolutionary Optimization, Constrained Optimization.

Self-Learning : Mathematical concepts of Objective function, Constraints, and Decision Variables.

Applications:

1. Minimize Loss Functions in Neural Networks using Batch Gradient Descent (Unconstrained Optimization).
2. Lagrange Multipliers to find Local Maxima and Minima of a function subject to equationconstraints (Constrained Optimization).
3. Case study on Bayesian Optimization with Discrete Variables (Discrete Variable Optimization).
4. Use Genetic Algorithms to Optimize Production Scheduling in a manufacturing environment, focusing on minimizing total production costs while meeting job deadlines and machine constraints.
5. Evaluate the GA's effectiveness against Traditional Scheduling Methods.

14 Hours

Tools / Languages/Libraries: Jupyter Notebook**,** Python, Pandas, Matplotlib, Scipy, Seaborn, BeautifulSoup, Numpy, Scikit learn.

Textbooks:

1: Statistics for Engineers and Scientists, William Navidi, McGrawHill Education,India,4thEdition, 2015.

2: Optimization Methods for Engineering Design, Parkinson, A.R., Balling, R., and J.D. Hedengren, Second Edition, Brigham Young University, 2018.

Reference Books:

1: The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modelling, Raj Jain, Wiley, 2008.

2: Sampling-DesignandAnalysis,SharonL.Lohr,2ndedition(stats),Cengage,2010.

3: Data Science from Scratch, Joel Grus,O Reilly,1stEdition,2015.

4: Statistics for Engineers and Scientists, William Navidi,3rdEdition,McGrawHill,2010.

UE23CS241A: Web Technologies (4-0-0-4-4)

Web Technologies course demonstrates an in-depth understanding of the technologies necessary for designing and developing a rich web application in an efficient way.

Course Objectives:

- Basic web technologies and building blocks of a website using HTML, CSS, JavaScript and Advanced JavaScript
- The core concepts of HTML5, JQuery and AJAX, MERN (MongoDB, ExpressJS, ReactJS and NodeJS) stack and build an UI of the application using React JS
- Building a multi-tier application by interfacing UI to NodeJS
- Integrate database MongoDB through Express JS Framework and Web services.

Course Outcomes:

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

- Understand basic web technologies like HTML, CSS and JavaScript
- Achieve rich user experience by implementing HTML5 features and Asynchronous communication using AJAX, JQuery and MERN stack layers (MongoDB, ExpressJS, ReactJS and NodeJS) and Create rich User Interface using React JS
- Understand and Integrate the UI with NodeJS
- Create RESTful Web services using ExpressJS and MongoDB database

Course Content:

Unit 1: HTML, CSS and Client Side Scripting
Introduction to Web Architecture and Web protocols (HTTP Request Response Formats, URLs), Basic Mark-ups & syntax, HTML elements & attributes, Web Form, HTML5 (New Tags, Inputs, Elements and Controls), CSS3.0 - Styles and Style sheets, Selectors, Style properties, Box Model, JavaScript Basics(variables, scope, Builtin Objects), JavaScript objects, DOM Manipulations, Events and Event Handling in JavaScript

14 Hours

Unit 2: HTML5, JQuery and Ajax
HTML5 (APIs), JQuery Introduction, Callbacks and Promises, Introduction to Single Page Application, XML Vs JSON, Asynchronous Communication using AJAX and fetch API. **ReactJS** – MERN Introduction, React Classes and Components, JSX, Rendering of elements

14 Hours

Unit 3: ReactJS
Properties, State, Context, Component lifecycle methods, Refs & Keys, Event Handling, Stateless components, React Forms, React Hook **NodeJS –** Understanding Node JS Architecture, Set up Node JS app, Node Modules, call-backs, File system Module, HTTP Module, Handling HTTP Requests

14 Hours

Unit 4: MongoDB
MongoDB-Documents, Collections, Reading and Writing to MongoDB, MongoDB NodeJS Driver, Running a react application on NodeJS, React Router. **ExpressJS –** Introduction to Web services and REST API's , Express Framework Overview, Routing and URL building, Error Handling, Express Middleware, Form Data and File Upload.

14 Hours

Tools / Languages: HTML, CSS, JavaScript, MERN Technologies.

Text Book(s) :
1: "Learning PHP, MySQL & JavaScript", Robin Nixon., 5th edition, O'Reilly Media, Inc. ISBN: 9781491978917, 2018.

2: “Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node”, Vasan Subramanian, Apress, 2017.

Reference Book(s) :

1: “Beginning Node.js, Express & MongoDB Development”, Greg Lim, July 2019.
2:“Learning React, Functional Web Development with React and Redux”, Alex Banks and Eve Porcello, O’Reilly Media, May 2017.

Reference Link(s);

1: https://reactjs.org/docs/.
2:https://www.kirupa.com/react.

UE23CS242A: Automata Formal Languages and Logic (4-0-0-4-4)

The course introduces fundamental concepts in Automata and Formal Languages and their application to Logic. Thecourse covers the notions of Finite State Automaton, Regular expression, Push down Automaton, Context FreeLanguages and Turing Machines. These abstract and formal models and their usage in Propositional and First Order Predicate Logic, allow for solving problems in Formal language Generation and Recognition.

Course Objectives:

- Teach students to construct basic machines like DFA, NFA which represent Regular Languages, Regular Expressions, and Regular Grammars and to identify Non – Regular Languages.
- Teach students to identify Context Free Languages, to construct Push down Automata which represent Context Free Languages, to convert the given grammar to various normal forms and to make use of Membership Algorithm.
- Teach students to understand closure properties of Context Free Languages, to identify Non – Context Free Languages and to construct Turing Machines and
- To familiarize students with concepts like Recursively Enumerable languages, Recursive Languages, Undecidable Problems. And to familiarize notions of mathematical logic: logical notations (syntax) and how to assign meaning to them (semantics).

Course Outcomes:

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

- Design simple machines like DFA, NFA, convert NFA to DFA and minimize a given DFA, construct regular expressions for different languages, verify that some languages are regular and some are not.
- Analyze the difference between Regular Languages and Context Free Languages, design Push Down automata,construct Context Free Grammars, and convert one form of the grammar to another form.
- Enumerate the properties of Context Free Grammars, verify that some languages are context free and some arenot, design Turing Machines, and analyze the difference between acceptability and decidability and
- Analyze the difference between Recursive and Recursively Enumerable Languages, Decidable Languages,Turing – Recognizable and Co – Turing – Recognizable, some problems that cannot be solved by Turing Machines, reduce one Undesirables Problem to another, Undeniable Problems for Recursively Enumerable Languages. And make use of Propositional Logic and Predicate Logic in knowledge representation and truth verification.

Course Content:

Unit 1: Introduction

Mathematical Preliminaries and Notation, Three Basic Concepts. Finite Automata: Deterministic Finite Accepters, Non-Deterministic Finite Accepters, Equivalence of Deterministic and Non-Deterministic Finite Accepters,Reduction of the number of states in Finite Automata. Regular Expressions, Connection between Regular Expressions and Regular Languages Regular Grammars.

14 Hours

Unit 2: Regular Languages and Context Free Languages

Properties of Regular Languages: Closure Properties of Regular Languages, Elementary Questions about Regular Languages, Identifying Non Regular Languages. Definitions of PDA and CFL, Deterministic Pushdown Automata,Non-Deterministic Pushdown Automata, Pushdown Automata and Context Free Languages, Context Free Grammars.

14 Hours

Unit 3: Properties of Context Free Languages and Turing Machine

Parsing and Ambiguity. Simplification of Context–Free Grammars and Normal Forms: Methods for Transforming Grammars, Two Important Normal Forms, A Membership algorithm for Context Free Grammar. Properties of Context-Free Languages: Closure Properties and Questions about Context–Free Languages, Pumping Lemma forContext–Free Languages. Turing Machines: The Standard Turing Machine, Constructing Turing Machines, Combining Turing Machines for Complicated Tasks, Turing's Thesis, Rice theorem.

14 Hours

Unit 4: Undecidability and design of a declarative language.

Hierarchy of Formal Languages and Automata: Recursive and Recursively Enumerable Languages, the ChomskyHierarchy. Limits of Algorithmic Computation: Some Problems that cannot be solved by Turing Machines,Undecidable Problem for Recursively Enumerable Languages, idea of reduction. A very simple Logic, Syntax, Semantics, A simple knowledge Base, A simple inference procedure. PropositionalTheorem Proving: Inference and Proofs, Proof by Resolution, Conjunctive Normal Form, A resolution algorithm.Syntax and Semantics of First Order Logic: Models for First Order Logic Symbols and interpretations, Terms,Atomic Sentences, Complex Sentences Quantifiers, Equality, Numbers, sets and Lists. Example - The electronic circuits' domain.

14 Hours

Tools / Languages: JFLAP.

Text Book(s):

1: “An Introduction to Formal Languages and Automata”, Peter Linz, Jones and Bartlett, New Delhi, India, 6th Edition, 2016.
2: “Artificial Intelligence – A Modern Approach”, Stuart Russell and Peter Norvig, Pearson, 3rd Edition (Paperback), 2016.

Reference Book(s):

1: “Theory of Computation”, Michael Sipser, Cengage Learning, New Delhi, India, 2008.
2: “Introduction to Automata Theory, Languages, and Computation”, John E Hopcroft, Rajeev Motwani, Jeffrey D Ullman, Pearson Education, New Delhi, India, 3rd Edition, 2009.
3: “Theory of Computation: A Problem–Solving Approach”, Kavi Mahesh, Wiley India, New Delhi, 2012.

UE23CS251B: Microprocessor and Computer Architecture (4-0-2-5-5)

This course will give you an in-depth understanding of the inner-workings of modern digital computer systems and trade-offs present at the hardware-software interface. The course focuses on key topics in microprocessor such as the system architecture, low level programming aspects and interface with other key components. Also, the course will help in understanding the core computer architecture concepts such as multilevel in memory hierarchies, pipelining, and super scalar techniques. A desirable knowledge of Digital Design and Computer organization is required.

Course Objectives:

- Introduce concepts of basic processor architecture and its design.
- Understanding the concept of concepts of pipeline architecture and hazards.
- Study of memory hierarchy, cache memory and its optimizations.
- Introduce advanced concepts in processor architecture like multi-core/ many core processor architectures.

Course Outcomes:

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

- Demonstrate ability to understand the design of different instruction sets like RISC/ CISC and their addressing modes.
- Demonstrate the ability to understand the design of a pipelined processor and its challenges.
- Demonstrate the use of tools to analyse the performance of programs on different architectures. Design alternative memory hierarchy layouts and optimizations.
- Demonstrate and appreciate modern trends in architecture such as multicore architectures.

Desirable Knowledge: UE23CS251A- Digital Design and Computer Organization.

Course Content:

Unit 1 : Architecture

Introduction, ISA Classification - RISC and CISC, Memory Addressing, Operands - Types and Size, Instruction Set - Operations, Control Flow, Instruction Encoding, Case Study - ARM/ MIPS/ x86 Processor.

14 Hours

Unit 2 : Pipelining

3- Stage Pipelining, 5 - Stage Pipelining, Pipeline Datapath and Control, Data Hazards – Forwarding vs. Stalling, Control Hazards, Branch Prediction Mechanisms and Exceptions, Performance Metrics.

14 Hours

Unit 3 : Basics of Cache and Cache Optimization

Basics of Caches - Fully Associative, Direct Mapped and Set Associativity, Cache Performance, Basic Cache Optimization- Reduce in Miss Rate. Basic Cache Optimization- Reduce Miss Penalty, Reduce Hit Time.

14 Hours

Unit 4 : Advances in Architecture

Introduction to Parallel Computing, PC – Applications, Memory architecture, Flynn's taxonomy, parallel programming models, Shared memory programming OpenMP-Introduction, loop-level parallelism, CUDA C Program structure-vector kernel addition, device global memory and Data transfer, Hardware Multi threading, Parallel examples: matrix multiplication, PC-Design Issues, Amdahl's Law, Gustafson Law, Multi-Core Architecture, Introduction to GPU computing

14 Hours

.

Laboratory :

Course Content:

1. Introduction to ARM Simulator and sample programs
2. Implementation of Data Processing Instructions, Programs on usage of addressing modes
3. Programs on functions and software interrupts, Matrix Operations – Addition and Multiplication
4. Introduction to PARACACHE simulator – Direct Mapping, Associative Mapping
5. Introduction to Plugins interface.
6. Project work

Tools/ Languages: ARM Simulator, Arduino Microcontroller kit, ParaCache simulator.

Text Book(s):

1: “Computer Organization and Design”, Patterson, Hennessey, 5th Edition, Morgan Kaufmann, 2014.
2. “Computer Organization and Design – ARM Edition”, Patterson, Hennessey, 4th Edition, Morgan Kaufmann, 2010.
3. “ARM System-on-Chip Architecture”, Steve Furber, 2nd Edition, Pearson India, 2015.
4. “Programming Massively Parallel Processors A Hands-on Approach”, Third Edition
David B. Kirk Wen-mei W. Hwu, Morgan Kaufmann
5. “Multicore and GPU Programming An Integrated Approach”, Gerassimos Barlas, second edition, Morgan Kaufmann

Reference Book(s):

1: “Computer Architecture: A Quantitative Approach”, Hennessey, Patterson, 5th Edition, Morgan Kaufmann, 2011.
2: “The Definitive Guide to the ARM Cortex-M0 and Cortex MO+ processors”, Joseph Yiu, 2nd Edition, Newnes, 2015

UE23CS252B: Computer Networks (4-0-2-5-5)

This is a foundation course on Computer Networking which focuses on building blocks of the Internet. We trace the journey of messages sent over the Internet from an application residing on one host machine (source) to another (Destination) using a top-down layered approach. The course contents are organized based on TCP/IP Protocol stack.

Course Objectives:

- To present a broad overview of computer networking, the Internet and network layered architecture.
- To study the conceptual and implementation aspects of network applications, protocols and socket programming.
- To provide an insight of the Internet's connection–oriented and connectionless end–to–end transport service protocols and TCP's approach to congestion control, to learn exactly how the network layer can provide its host–to–host communication service.
- To study the IPv4 and IPv6 protocols, to explore several important link–layer concepts and technologies, LAN and Wireless LANs.

Course Outcomes:

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

- Demonstrate in a concise way how the Internet is constructed and functions with respect to TCP/IP or OSI reference models.
- Demonstrate application layer protocols like DNS, HTTP, HTTPS and be able to develop simple client–server applications using socket programming and understand the concept of unreliable data transfer protocols and how UDP implement these concepts.
- Understand the concept of reliable data transfer protocols and how TCP implement these concepts. Implement logical addressing schemes and configure devices using NAT.
- Demonstrate the ability to configure the routers and services such as DHCP, ICMP. Construct and troubleshoot a wired or wireless LAN, and be able to understand wider networking issues.

Course Content:

Unit 1- Computer Networks and the Internet, Application Layer

Introduction to Computer Networks, Internet: A Nuts–and–Bolts Description, A Services Description, Protocol, The Network Edge: Access Networks, Physical Media, Introduction to physical layer devices, The Network Core, Packet Switching, Circuit Switching, A Network of Networks, Delay, Loss, and Throughput in Packet–Switched Networks, Overview of Delay in Packet Switched Networks – Queuing Delay and Packet Loss, End–to–End Delay, Throughput in Computer Networks, The OSI Model and the TCP/IP Protocol Suite, Protocol Layers, The OSI Model, TCP/IP Protocol Suite. Network Application Principles: Network Application Architectures, Processes Communication, Transport Services available to Applications, Transport Services provided by Internet.

14 Hours

Unit 2 - Application Layer, Transport Layer – UDP

The Web, HTTP and HTTPS, Non persistent and persistent connection, HTTP Message Format, User Server Interaction: Cookies, Web Caching. DNS, The Internet's Directory Service: Services provided by DNS; How DNS works: DNS Records and messages; Peer to peer Applications; Socket Programming with TCP and UDP; Other Application Layer Protocols: FTP, SMTP, SNMP, Telnet, SSH.

Introduction to Transport Layer Services: Relationship Between Transport and Network Layer, Overview of the Transport layer in the Internet, Multiplexing and Demultiplexing; Connectionless Transport UDP: UDP Segment Structure, UDP Checksum.

14 Hours

Unit 3- Transport Layer – TCP, Network Layer and Internet Protocol

Principles of Reliable Data Transfer: Building a Reliable Data Transfer Protocol, Pipelined Reliable Data Transfer Protocol, Go–Back–N Protocol, Selective–Repeat; Connection Oriented Transport TCP: The TCP Connection, TCP Segment Structure, Flow Control, TCP Connection Management, TCP Congestion Control. Numerical on TCP congestion control mechanisms–TCP Tahoe, Reno.

Overview of Network Layer: Forwarding and routing, what's Inside a Router? The Internet Protocol (IP) IPV4: Datagram Format, Fragmentation, Addressing, NAT.

14 Hours

Unit 4 - Network Layer and Internet Protocol, Link Layer and LAN

Introduction to Network layer Protocols: DHCP, ICMP; IPv6 Protocol: Packet Format, Transition from IPv4 to IPv6; Introduction to Routing Algorithms: Link State: Dijkstra's algorithm and Distance Vector: Bellman–Ford Algorithm.
Link layer – Error–Detection and Correction techniques, Parity checks, Internet Checksum, Cyclic Redundancy Check, and Multiple Access Protocols: CSMA/CD, CSMA/CA; Switched LAN: Link layer addressing and ARP, Ethernet: Link–layer switches. Retrospective: A Day in the Life of a Web Page Request. Physical Layer – Purpose, Signals to Packets, Transmission media. Wireless LANs: IEEE 802.11 LAN architecture, 802.11 MAC Protocol, IEEE 802.11 Frame.

14 Hours

Tools: Wireshark, Python.

Text Book(s):

1: "Computer Networking: A Top – Down Approach", James F. Kurose, Keith W. Ross, 7th Edition, Pearson Publication, 2017.

Reference Book(s):

1: "TCP IP Protocol Suite", Behrouz Forouzan, 4th Edition, McGraw–Hill, 2010.

Lab/ Hands-on: **14 Hours**

1. Program on ping, tcpdump and wireshark.
2. Program on Exploring HTTP with wireshark, Web Server setup, FTP/ SMTP and SNMP Clients, Telnet, SSH and DNS
3. Program on Wireshark based TCP congestion window plotting, UDP traffic analysis.
4. Program on Cisco Packet Tracer based Router experiments; IPv4 Fragmentation based Wireshark experiments, Inspection of DHCP, ICMP.
5. Program on Wireshark based Link Layer protocol inspection.

UE23CS241B - Design and Analysis of Algorithms (4-0-0-4-4)

Algorithms play a key role in science and practice of computing. Learning algorithm design technique is a valuable endeavour from practical standpoint and algorithm design techniques have considerable utility as general problem solving strategies, applicable to problems beyond computing. This course includes classic problems of computer science, application of design techniques and analysis in terms of time and space.

Course Objectives:

- Learn various algorithm design techniques and apply appropriate algorithmic design techniques for specific problems.
- Learn to design and analyze algorithms with an emphasis on resource utilization in terms of time and space
- Learn to trade space for time in algorithmic design using input enhancement and per-structuring.
- Learn the limitations of algorithmic power and techniques to handle these limitations

Course Outcomes:

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

- Identify the design technique used in an algorithm
- Design and implement efficient algorithms for practical and unseen problems and analyze these algorithms using quantitative evaluation.
- Analyse time efficiency over trading space
- Understand the limits of algorithms and the ways to cope with the limitations.

Course Content:

Unit 1: Introduction and Brute Force

Algorithms, Fundamentals of Algorithmic Problem Solving, Important Problem Types. Analysis of Algorithm Efficiency: Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non Recursive and Recursive Algorithms. Brute Force: Selection Sort, Bubble Sort, Sequential Search, Brute Force String Matching, Exhaustive Search.

14 Hours

Unit 2: Decrease – and – Conquer & Divide-and-Conquer

Decrease-and-Conquer: Decrease by constant number algorithms - Insertion Sort, Topological Sorting, Algorithms for Generating Combinatorial Objects, Decrease-by-a-Constant-Factor Algorithms – Fake Coin Problem, Russian Peasant Multiplication, Josephus problem, Decrease-by-Variable-Size Algorithms – Computing a median and the selection problem. Divide-and-Conquer: Master Theorem, Merge Sort, Quick Sort, Binary Search, Binary Tree Traversals, Complexity analysis for finding the height of BST, Multiplication of Large Integers, Strassen's Matrix Multiplication.

14 Hours

Unit 3: Transform-and-Conquer Space and Time Tradeoffs & Greedy Technique

Transform and Conquer**:** Pre-sorting, Heap Sort, Red-Black Tree Construction and Time complexity Analysis for insert and search operation, 2-3 Trees and B Tree: insertion, deletion, searching, and time complexity analysis. Space and Time Tradeoffs: Sorting by Counting, Input Enhancement in String Matching - Horspool's and Boyer-Moore Algorithms. Greedy Technique: Prim's Algorithm, Kruskal's Algorithm and union-find algorithm, Dijkstra's Algorithm, Huffman Trees

14 Hours

Unit 4: Limitations, Coping with the Limitations of Algorithm Power & Dynamic Programming,

Dynamic Programming: Computing a Binomial Coefficient, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms. Limitations of Algorithm Power**:** Lower-Bound Arguments, Decision Trees, P, NP, and NP-Complete, NP-Hard Problems. Coping with the Limitations of Algorithm Power: Backtracking, Branch-and-Bound.

14 Hours

Tools / Languages: C Programming Language, GCC Compiler.

Text Book(s):

1: "Introduction to the Design and Analysis of Algorithms", Anany Levitin, Pearson Education, Delhi (Indian Version), 3rd Edition, 2012.

Reference Book(s):

1: “Introduction to Algorithms”, Thomas H Cormen, Charles E Leiserson, Ronald L Rivest and Clifford Stein, Prentice-Hall India, 3rd Edition, 2009.
2: “Fundamentals of Computer Algorithms”, Horowitz, Sahni, Rajasekaran, Universities Press, 2nd Edition, 2007.
3: “Algorithm Design”, Jon Kleinberg, Eva Tardos, Pearson Education, 1st Edition, 2006.

UE23CS242B: Operating Systems (4-0-0-4-4)

This course focuses on fundamental operating systems concepts including various algorithms and trade-offs for efficient management of resources such as CPU, Memory, Storage and I/O. This course requires the student to have a desirable knowledge of Data Structures and its Applications.

Course Objectives:

- Learn the fundamental Operating System concepts and various algorithms for process management.
- Demonstrate various algorithms and associated trade-offs for efficient resource management such as inter process communication, threads, process synchronization, deadlocks.
- Provide an understanding and apply various memory management techniques.
- Understand file system and secondary storage structures.

Course Outcomes:

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

- Understand the design of various algorithms for scheduling and their relative performance.
- Understand and apply Inter Process Communication, threads, process synchronization and deadlocks
- Apply various memory management techniques.
- Understand file system, its implementation and the various secondary storage structures.

Desirable Knowledge: UE23CS252A - Data Structures and its Applications.

Course Contents:

Unit 1 : Introduction and Process Management

What Operating Systems Do, Operating-System Structure & Operations, Kernel Data Structures, Operating-System Services, Operating System Design and Implementation.

Shell programming: Overview of bash shell programming – variables, control flow

Processes: process concept, Process Scheduling, Operations on Processes, System calls for process management-fork (), vfork (), wait () and exec ().

CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms. Case Study: Linux Scheduling Policies.

Shell programming – cron.

14 Hours

Unit 2 : IPC, Threads and Concurrency

IPC: Introduction, Shared Memory systems, Message Passing, Communication in Client–Server Systems- Pipes, ordinary pipes and named pipes, system calls for shared memory, pipes and FIFO's.

Threads: Overview, Multicore Programming, Multithreading Models, Thread Libraries, Thread Scheduling.

Process Synchronization: Background, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization- The Bounded-Buffer Problem, The Readers–Writers Problem, The Dining-Philosophers Problem, Synchronization Examples- Synchronization in Linux. System calls for threads creation and synchronization-POSIX Threads.

Deadlocks: System Model, Deadlock Characterization, Deadlock avoidance, Banker's Algorithm, Deadlock Detection, Deadlock Recovery.

14 Hours

Unit 3: Memory Management

Main Memory: Background- Basic Hardware, Address Binding, Logical Versus Physical Address Space, Dynamic Loading, Dynamic Linking and Shared Libraries, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table.

Virtual Memory: Background, Demand Paging, Copy-on-Write, Page Replacement Algorithms-FIFO, LRU, Optimal, Allocation of Frames, Thrashing.

14 Hours

Unit 4 : File and Storage Management

File-System Interface: File Concept, system calls for file operations-open(), read(),write(), lseek(), close() and system call to retrieve file attributes and file types-stat(), lstat(), Access Methods, Directory and Disk Structure, system calls for reading directories, system calls to create hard links (link()) and symbolic links-symlink().

File-System Implementation: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, File Sharing, Protection.

Storage management: Overview of Mass-Storage Structure, Disk Scheduling, Swap-Space Management, RAID Structure.

System Protection: Goals, Principles and Domain of Protection, Access Matrix, Implementation of the Access Matrix, Access Control; **System Security**
Shell programming - awk, sed

14 Hours

Tools/Languages/OS : C, Linux/Unix OS for system call implementation.

Text Book(s):
1: "Operating System Concepts", Abraham Silberschatz, Peter Baer Galvin, Greg Gagne 9th Edition, John Wiley & Sons, India Edition ,2016.
2: "Advanced Programming in the Unix Environment", Richard Stevens and Stephen A Rago, Pearson, 3rd edition, 2017.
3: "Your UNIX/Linux: The Ultimate Guide", Sumitabha Das, 3rd Edition, McGraw-Hill, 2013.

Reference Book(s):
1: "Operating Systems, Internals and Design Principles", William Stallings, 9th Edition, Pearson, 2018.
2: "Modern Operating Systems", Andrew S Tanenbaum, 3rd edition, Pearson, 2007.
3: "Learning the bash shell", Cameron Newham, 3rd edition, O'Reilly, 2005

UE23MA241B: Linear Algebra and Its Applications - (4-0-0-4-4)

This is a basic subject on matrix theory and linear algebra. Emphasis is given to topics that will be useful in Engineering discipline, including system of equations, vector spaces, eigenvalues, similarity, and positive definite matrices.The course provides hands-on experience in basic programming concepts using PYTHON/MATLAB for solving problems relevant to these areas.

Course Objectives:

- To teach students how to use linear algebra as a powerful tool for computation.
- To show how these computations can be conceptualized in a geometric framework.
- To give a gentle introduction to the theory of abstract vector spaces.
- To visualize solution to linear system of equations with different approaches using Python/Matlab.

Course Outcomes:

After completing this course, students will be able to:

- Solve systems of Linear Equations using Matrix Transformations, Interpret the nature of Solutions, Visualize Consistency of Linear system of Equations and also compute inverse of a Matrix.
- Demonstrate the ability to work within Vector Spaces, distil Vector Space properties and understand the concepts of the four fundamental Subspaces, Linear Span, Linear Independence, Dimension and Basis
- Learn the concepts of Orthogonal Vectors and Orthogonal Subspaces and apply the Gram-Schmidt process to find an Orthonormal Basis in a Subspace, Eigenvalues, Eigenvectors and Diagonalization of a Matrix.
- Apply the concept of Positive Definite Matrices, Singular Value Decomposition into application problems.

Course Content:

Unit 1: Matrices, Gaussian Elimination and Vector Spaces

Introduction, The Geometry of Linear Equations, Gaussian Elimination, Singular Cases, Elimination Matrices, Triangular Factors -LU decomposition & Cholesky's method and Row Exchanges, Inverses and Transposes, Inverse by Gauss -Jordan method, Vector Spaces and Subspaces (definitions only)
Self-Learning Component: Algebra of Matrices.

14 Hours

Unit 2: Four Fundamental Subspaces & Linear Transformations

Linear Independence, Basis and Dimensions, Row reduced Echelon form, The Four Fundamental Subspaces, Rank-Nullity theorem. Linear Transformations, Algebra of Linear transformations,
Self-Learning Component: Examples of Vector Spaces and Subspaces.

14 Hours

Unit 3: Orthogonalization, Eigenvalues and Eigen vectors

Orthogonal Vectors and Subspaces, Orthogonal Bases, Cosines and Projections onto Lines, Projections and Least Squares. Orthogonalization, The Gram-Schmidt Orthogonalization process, Introduction to Eigenvalues and Eigenvectors, Properties of Eigenvalues and Eigenvectors, Cayley-Hamilton theorem (statement only), Diagonalization of a Matrix.

14 Hours

Unit 4: Singular Value Decomposition

Symmetric Matrices, Quadratic Forms, Definitions of Positive definite, negative definite, positive semi-definite, negative semi-definite, indefinite forms and matrices, Tests for Positive Definiteness, Singular Values and Singular Vectros, Image Processing by Linear Algebra, Principal Component Analysis (PCA by the SVD), Minimizing a Multivariate Fuctions, Back Propagation and Stochastic Gradient Descent.

14 Hours

Text Books:

1: Linear Algebra and its Applications, Gilbert Strang, Thomson Brooks/ Cole, 5th Edition, Second Indian Reprint 2007.
2: Application : Applied projects for an introductory linear algebra class by Anna Zemlyanova.

Reference Books:

1: Linear Algebra and its Applications, David. C lay, Publication by Pearson Education, 5th Edition, 2015
2: Linear Algebra, Schaum's outlines, Seymour Lipschutz and Marc Lipson, Tata McGraw-Hill publications, 4th Edition, 2009.
3: Higher Engineering Mathematics, B S Grewal, Khanna Publishers, 44th Edition, 2020,.
4: Practical Linear Algebra, Gerald Farin and Dianne Hansford, CRC Press, Taylor & Francis Group, 3rd Edition.

Hands on learning using PYTHON and Applications to Computer Science Engineers.

Unit 1: Basic operations with matrices in Matlab/Python.
Matrix operations and image manipulation
Matrix multiplication, inversion, and photo filters
Solving linear systems

Unit 2:
Systems of linear equations and college football team ranking (with an example of the Big 12) Convolution, inner product, and image processing revisited
Norms, angles, and your movie choices
Interpolation, extrapolation, and climate change

Unit 3:
Orthogonal matrices and 3D graphics
Discrete dynamical systems, linear transformations of the plane, and the Chaos Game
Projections, eigenvectors
Matrix eigenvalues and the Google's PageRank algorithm

Unit 4:
Principal Component Analysis, and face recognition algorithms
Social networks, clustering, and eigenvalue problems
Singular Value Decomposition and image compression

UE22CS351A: Database Management System (5-0-2-5-5)

This course offers a solid theoretical foundation in Database Management System (DBMS) and explores the practical applications of DBMS in a real-world scenario. The course emphasizes on the creation and the design of relational database systems. It also introduces the databases that provides flexible schemas and scale easily with large amounts of data and high user loads.

Course Objectives:

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

- Understand fundamental concepts, terminology, and application of relational databases and Construct ER diagrams for a desired application and transform the same into a relational schema
- Understand database design concepts and design relational databases and construct basic and advanced SQL queries
- Understand, and apply Normal Forms in database design, Transactions, Concurrency control, Locking, and Recovery, implement them in a real-time setup, and transform the given Normal Form into the desired form.
- Understand the concepts of Database Security, and NoSQL databases such as MongoDB, Neo4j, and DynamoDB

Course Outcomes:

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

- Demonstrate an ability to explain the basic concepts of database management, construct and transform ER diagrams into Relational Schema.
- Design databases and construct simple and advanced SQL queries for given contexts.
- Explain Normal Forms, employ them in Database Design, and apply database security concepts in application contexts.
- Demonstrate the ability to use semi structured and NoSQL databases

Course Content:

Unit 1: Introduction to Database Management

Database System Applications, Purpose, View of data, Database Languages, Database design- Introduction to databases, Database application architecture, Users and Administrators, E-R Model, reducing ER to a relational schema. Structure of relational databases, Database schema, and its constraints, Keys

14 Hours

Unit 2: Relational Model and Database Design

Relational operations (Algebra), Unary Operations - Unity, Binary, Aggregate Functions, Grouping, SQL overview, Data definition, Structure of SQL queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Subqueries, Database Modification, Join expressions, Views, Triggers, Functions, and Procedures, Introduction to strategies of Query processing and Query optimization.

14 Hours

Unit 3: Advanced Design Concepts and Implementation

Functional Dependencies, Inference Rules, Closure, Equivalence, Minimal Cover Normal Forms Based on Primary Keys (1NF, 2NF, and 3NF), General Definitions of Second and Third Normal Forms Boyce-Codd Normal Form, Properties of Relational Decompositions, Overview of Higher Normal Forms.

14 Hours

Unit 4: Advanced Databases

Database transactions, Concurrency control, Locking, Recovery, Database Security,Introduction to NoSQL databases, CAP theorem, Document database (MongoDB), Key-Value database (DynamoDB), Graph databases (Neo4j),Wide-column store database(HBase)

14 Hours

Tools/ Languages: MySQL Workbench, Python, ERwin, Any other tool for ER modeling

Lab/Hands-on sessions **14 Hours**

1. Draw an ER diagram for a given problem statement
2. Conversion of an ER diagram into Relational schema

3. DDL – create, constraints, alter, rename, drop, truncate table, Views.
4. DML – Insert, Update, Delete, Transactions - commit, rollback, savepoint
5. SQL - Set operators: union, intersect, minus.
6. SQL – Aggregate functions.
7. SQL – Joins: inner, outer; Sub queries: correlated and uncorrelated
8. SQL – Creating Functions and Procedures
9. SQL – Creating Triggers and Cursors
10. XML- Database access
11. NoSQL database queries
12. High-level programming language accessing a database using an API.

Text Book(s):

1: “Fundamentals of Database Systems”, Ramez Elamsri, Shamkant B Navathe, Pearson, 7th Edition, 2017. 2: “Database System Concepts”, Silberschatz, H Korth and S Sudarshan, McGrawHill, 7th Edition, 2019.

Reference Book(s):

1: Database Management Systems, R Ramakrishnan, J Gehrke, 3rd Edition, McGraw Hill, 2002

2: Data on the Web: From Relations to Semistructured Data and XML, S Abiteboul, P Buneman, D Suciu, Morgan Kauffman, 1999

UE22CS352A: Machine Learning (4-0-2-5-5)

Machine Learning surrounds us today: in phones that respond to voice commands, programs that beat humans at Chess and Go, robots that assist surgeries, vehicles that drive in urban traffic, and systems that recommend products to customers on e-commerce platforms. This course aims to familiarize students with the breadth of modern ML, to impart an understanding of the dramatic surge of ML in the last decade, and to foster an appreciation for the distinctive role that ML can play in shaping the future of our society. This course requires the student to have a desirable knowledge of Statistics for Data Science, Linear Algebra and its Applications and Design and Analysis of Algorithms.

Course Objectives:

- Formulate a well-defined Machine Learning problem with clear metrics.
- Understand the notions of Hypotheses Space, Hypotheses Structure and Search.
- Become conversant with types of Machine Learning Algorithms, their applicability and Inductive Bias.
- Familiarize with techniques for Ensemble Learning and deep learning.

Course Outcomes:

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

- Distinguish categories of Data Attributes, Dimensions, and Sample Sizes.
- Acquire a thorough understanding of Supervised, Unsupervised Learning,
- Apply Ensemble Methods and deep learning methods for solving the problems.
- Design and implement a solution to real world machine learning problem.

Desirable Knowledge: UE22CS241A - Statistics for Data Science, UE22MA241B- Linear Algebra & its Applications, UE22CS242B –Design and Analysis of Algorithms.

Course Content:

Unit1: Introduction, Performance Metrics, Classification with Decision trees and KNN: Introduction to ML: Machine Learning and its Models, Concept Learning: Concepts of hypotheses, Version space, inductive bias, Performance metrics-accuracy, precision, recall, sensitivity, specificity, AUC, ROC, Bias Variance decomposition. Decision Trees-Basic algorithm(ID3) for classification, Decision boundary for decision trees(x-y axis), Hypothesis search and Inductive bias, Issues in Decision Tree Learning – Over fitting, Solutions to over fitting, dealing with continuous values, Linear Regression with GD, Logistic Regression, Instance-based learning: k-nearest neighbor learning (Classification & Regression), Decision boundary for KNN.

14 Hours

Unit2: Ensemble Models, Supervised Learning with ANN and SVM: Combining weak learners, Improving performance with Gradient Boost, Random Forest, Artificial Neural networks: Introduction, Perceptrons, Multi-layer networks and back-propagation, Activation functions (Step, Sigmoid, Tanh, ReLU), Various Optimizers(GD, SGD, Momentum-based, Adagrad, Adam,), Support Vector Machines – margin and maximization, SVM - The primal problem, the Lagrangian dual, SVM –Solution to the Lagrangian dual Hard Margin(Classification ONLY)

Self Learning :Soft Margin(Classification ONLY), Kernel functions :Linear, polynomial (Derivation only for linear function).

14 Hours

Unit3: Stochastic Models, unsupervised learning : Bayesian Learning –Bayes theorem, Maximum likelihood, Bayes optimal classifier, Naïve Bayes classifier , Expectation maximization and Gaussian Mixture Models, Hidden Markov Models, Hierarchical vs. nonhierarchical clustering, Agglomerative and divisive clustering, K-means clustering, Bisecting K-means, K-Means as special case of Expectation Maximization.

14 Hours

Unit4 : Introduction to Deep Learning techniques and Transformers Introduction to Deep Learning, Introduction to Convolution operation and Convolution Neural Network. (Parameter calculation, Max pooling, Avg pooling).

Introduction to Recurrent Neural Network, Vanishing and exploding gradient, Basic GAN and its architecture, Basic architecture of Transformers- (Encoder-Decoder perspective with basics of attention mechanism)

Self learning : Variants of RNN: LSTM, GRU (mention of gates).

14 Hours

Tools/ Languages: Pytorch, Sk Learn, Keras, Tensor flow

Lab/ Hands-on:

o Decision Trees.
o ANN
o SVM
o Naïve Bayes
o HMM

o Clustering
o CNN model.

Text Books(s):

1: “Introduction to Machine Learning”, Ethem Alpaydin, Third edition, MIT Press Ltd.
2: “Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, The MIT Press, 2017.

Reference Book(s):

1:“Machine Learning”, Tom Mitchell, McGraw Hill Education (India),2013.
2: “Machine Learning: The Art and Science of Algorithms that Make Sense of Data”, Peter Flach, Cambridge University Press (2012).

UE22CS341A: Software Engineering (4-0-0-4-4)

Software Engineering course deals with the Software development life cycles, their individual phases, principles, methods, procedures and tools associated. This also deals with making of choices, implications of making choices and exposes students to the Software eco-system which will be experienced by students in a post college environment.

Course Objectives:

- Learn Software Engineering concepts. Ensure the relevance and need of an engineering approach to software development.
- Expose students to the software project management and Life Cycles.
- Enable the students to understand the principles of Software Configuration Management and importance of software quality for Software Product Development.
- Enable students to understand the continuous development, build, test and release of software products.

Course Outcomes:

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

- Relate to the challenges of Software Development and Software Engineering as a methodical approach for development.
- Able to Use Software project management ideas and apply suitable architecture and design approaches for software development
- Able to implement Software Configuration Management principles and evaluate software quality using quality metrics
- Work on a project plan, track and manage projects to Understand the connectivity of the development process to operations, and relate to the DevOps activities.

Course Content:

Unit 1: Introduction to Software Engineering and Requirements Engineering

Introduction, Context and Drivers of Software Engineering, (References: IEEE SWEBOK, PMBOK, BABOK and Other Sources from Internet) Processes Phases and Development lifecycle, Product Lifecycle, Legacy SDLCs - Waterfall Model, V Model, Incremental Model, Spiral Model, Evolutionary Model; 4Ps=(Process, Product, Project & People) Agile approach of software development, Contrasting Agile and Plan driven approaches, Agile SCRUM model and exposure to other Agile approaches like Lean Agile, Reuse focused Software Development approaches- CBSE, Product Line. Requirements Engineering: Requirement‟s introduction and properties, Feasibility, Requirements Elicitation, Analysis and modelling, Specification and Verification, Requirement Management and Requirements Traceability.

14 Hours

Unit 2: Software Project Management and Software Architecture and Design

Fundamentals, Software Project Management Lifecycle, Planning activities - Choice of Lifecycles, Project Organization, WBS, Software Estimation, Scheduling, Risk Management, Quality Management, Project Plan illustration using Microsoft Project or similar tool-, Gantt charts – Basic explanation Monitoring of Execution and Control, Project Closure. RACI matrix with example - ref to be checked Software Architecture and Design: Software Architecture, Characteristics and factors, Architectural approach including decomposition, Design and enabling techniques, key issues to be addressed as part of design, Architectural choices, impacts and conflicts, Generic Design approach, Architectural View, Styles, Architectural and Design Patterns. Contrast procedural and Object Orientation in Architecture and Design, Service Oriented Architecture (SOA).

14 Hours

Unit 3: Implementation, SCM and Software Quality & Relationship to Testing

Introduction to software construction, Coding principles, standards and guidelines, Factors for effective coding -Defensive, Secure and Testable programming, Code Review / Peer Review, Managing Construction and exposure to tools. Software Configuration Management (SCM): Elements of a Configuration Management System, Configuration Items, Baselines, Repositories, Branch Management, Build Management, Install, Change and Release Management, Patching and Patch Management, Configuration Management Plan – Relate to tools like GitHub. Introduction to Software Quality: Software Quality, A Taxonomy of Quality Attributes, Perspectives on Quality, Cost of Quality, Metrics, Software Quality Assurance, SEI CMM, Testing and Quality-

14 Hours

Unit 4: Software Testing, Maintenance and Software Ethics

Fundamentals of testing, Test Objectives, Verification and Validation, Terminologies, Testing Types – Black box, White box, Gray box, Static and Dynamic testing, Technique based testing – Coverage based, Fault based, Levels of testing – Unit testing, Integration testing, System testing and Acceptance testing, Manual and Automated testing,

Test Planning: test adequacy criteria, test strategy, test models, test schedule, test resource management, milestones, risks and measure and test plan. Test roles and responsibilities, test process/lifecycle Test case generation, test execution and debugging. Test metrics. Exposure to test tools like Junit, Selenium.

Software Maintenance, Lifecycle, activities and techniques. SE in Global Environment, and DevOps, DevSecOps CI/CD pipeline.

14 Hours

Tools / Languages: GitHub, MS Project/GanttPro/Jira, Jenkins.

Text Book(s):

1: “Software Engineering: A Practitioner’s Approach”, Roger S Pressman, McGraw Hill, 9th Edition, 2019
2: “Software Testing – Principles and Practices”, Srinivasan Desikan and Gopalaswamy Ramesh, Pearson, 2006.

Reference Book(s):

1: “Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling” by By Jennifer Davis, Ryn Daniels, O’ Reilly Publications, 2018
2. “Software Engineering: Principles and Practice", Hans van Vliet, Wiley India, 3rd Edition, 2010.
3: “Software Engineering”, International Computer Science Series, Ian Somerville, Pearson Education, 10th Edition, 2021
4: “Foundations of Software Testing ", Aditya Mathur, Pearson, 2008.
5: “Software Testing, A Craftsman’s Approach ", Paul C. Jorgensen, Auerbach, 2008.
6:IEEE SWEBOK, PMBOK, BABOK and Other Sources from Internet

UE22CS342AA1: Advanced Algorithms (4-0-0-4-4)

Algorithm Design and Analysis is fundamental and important part of computer science. The course on Advanced Algorithms introduces the learner to advanced techniques for design and analysis of algorithms and explores a variety of applications. This course requires the student to have a desirable knowledge of Design and Analysis of Algorithm.

Course Objectives:

- Enable the learner with basics of Amortized Complexity Analysis of Data Structures.
- Empower the learner with String Matching/ Prediction Algorithms and hone the problem solving skills of the learner by introducing them to Flow Networks, Bipartite Matching and DFT.
- Introduce the learner to Number Theoretic Algorithms.
- Enable the learner with design strategy of Dynamic Programming, Randomized Algorithms and Approximation Algorithms.

Course Outcomes:

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

- Perform Amortized Analysis of complex Data Structures.
- Apply String Matching Algorithms to solve string related problems and implement Max Flow and FFT Algorithms.
- Apply Number Theoretic concepts in designing Cryptographic Algorithms.
- Solve complex problems using Dynamic Programming, Randomized Algorithms and Approximate Algorithms.

Desirable Knowledge: UE22CS241B – Design and Analysis of Algorithms.

Course Content:

Unit 1: Basics of Complexity

Asymptotic Notations, Standard notations and common functions, Recurrences and Solution of Recurrence equations- The Substitution method, the Recurrence tree method, the Master method, Amortized Complexity Analysis: Aggregate, Accounting and Potential Methods.NP-Completeness, NP Reduction.

14 Hours

Unit 2: String Matching Algorithms, Suffix Trees and Maximum Flow,

Naive String Match, Boyer–Moore, Rabin–Karp, String matching with Finite State Automata, and Knuth–Morris–Pratt Algorithms Suffix Trees - Applications of Suffix Trees, Regular Expression Searches Using Suffix Trees.
Flow Networks, The Ford-Fulkerson method, The Edmonds-Karp algorithm, Maximum Bi-Partite Matching,

14 Hours

Unit 3: Polynomials and FFT &Number-Theoretic Algorithms

Polynomials and FFT: Representation of Polynomials, Efficient Polynomial Multiplication, DFT and FFT, Efficient Implementation of FFT.
Number-Theoretic Algorithms**:** Elementary notions; GCD, Modular Arithmetic, Solving modular linear equations, Modular Inverse, The Chinese remainder theorem, Powers of an element; RSA cryptosystem; Primality testing; Integer factorization.

14 Hours

Unit 4: Dynamic Programming Randomized Algorithms and Approximation Algorithms

Elements of Dynamic Programming, Problems - Coin- Row, Rod-Cutting, Matrix-Chain Multiplication, Longest Common Subsequence. Randomized Algorithms: Hiring Problem, Indicator random variables, Approximation Algorithm: Vertex Cover Problem, TSP, The Subset Sum Problem, Linear Programming.

14 Hours

Tools / Languages: C or C++ Programming Language.

Text Book(s):

1: “Introduction to Algorithms”, T H Cormen, C E Leiserson, R L Rivest and C Stein, PHI, 3rd Edition, 2010.

Reference Book(s):

1:“The Algorithm Manual”, Steven Skiena, Springer, ISBN: 9788184898651, 2nd Edition, Springer, 2008.
2: “Randomized Algorithms”, R Motwani and P Raghavan, Cambridge University Press, 2011

UE22CS342AA2: Data Analytics (4-0-0-4-4)

The course explores the data analytics lifecycle: question formulation, data collection and cleaning, exploratory, analysis, visualization, statistical inference, prediction, and decision-making. Focuses on building analytical models using key principles and techniques. This course requires the student to have a desirable knowledge of Statistics for Data Science.

Course Objectives:
The objective(s) of this course is to,

- Assess a dataset for the type of data, inter-relationships between features and interpret the data meaningfully
- Understand the assumptions that underlie regression models and determine the appropriate model and parameters for a problem
- Recognize patterns such as trend and seasonality in a time series data and design appropriate forecasting models
- Understand the nuances of recommender systems, build and validate appropriate models for recommendations.

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

- Perform exploratory data analysis on a given set of data and identify/ interpret correlation between features.
- Narrow down a subset of regression techniques to model data and evaluate the efficacy of these models.
- Analyze a time series and build appropriate time series models for prediction and evaluate them.
- Analyze data to infer underlying patterns, select the appropriate techniques and formulate recommendations.

Desirable Pre-requisite: UE22CS241A – Statistics for Data Science

Course Content:

Unit 1: Exploratory data analysis, ANOVA and correlation analysis

Introduction to business analytics, review of descriptive analysis and dimensionality reduction, ANOVA, correlation analysis, hidden variables

14 hours

Unit 2: Regression Analysis

Simple linear regression (alternative computational approaches and an emphasis on model diagnostics and validation), multiple linear regression (alternative computational approaches and an emphasis on model interpretation, diagnostics and validation) and variations (such as ridge regression and lasso regression), an introduction to multivariate and nonlinear regression, logistic regression (emphasis on interpretation, model diagnostics and validation)

14 Hours.

Unit 3: Time Series Analysis and Markov Chains

Time series data components, moving average and exponential smoothing methods, regression models for forecasting, AR model identification using ACF/ PACF, concept of stationarity – its importance, testing for stationarity and converting a non-stationary signal to stationary, ARMA and ARIMA modelling, Discrete Markov chains, Markov chains with Absorbing States, expected duration to reach a state from other states, computing the retention probability and customer lifetime value, confounding variables.

14 Hours

Unit 4: Recommender Systems

Collaborative filtering, knowledge-based filtering: constraint-based and case-based (with a mention of how traditional supervised learning approaches and unsupervised learning such as, DBSCAN, can help arrive at recommendations), a brief introduction to text classification/ clustering for content-based filtering, ensemble methods (bagging and boosting - how they serve to improve upon results from traditional approaches), interpreting ML models, mining of association rules and evaluation of recommender systems, A/B testing.

14 Hours

Tools & Languages: R andPython.

Text Book:

1. "Business Analytics, The Science of Data-Driven Decision Making", U. Dinesh Kumar, Wiley 2022, second edition.
2. "Recommender Systems: The Textbook", Charu C. Agarwal, Springer 2016.

Reference Book(s):

1. Data Mining: Concepts and Techniques by Jiawei Han, Micheline Kamber and Jianei, The Morgan Kaufmann Series in Data Management Systems, Elsever publications, 3rd Edition, 2012.
2. The Elements of Statistical Learning, Trevor Friedman, Robert Tibshirani and Jerome Hastie, Data Mining, Inference and Prediction, Springer 2001.
3. Practical Data Science with R, Nina Zumel and John Mount, Manning Publications, 2014.

UE22CS342AA3 : Internet of Things (4-0-0-0-4)

The Internet of Things is already changing the way people live and interact with humans and machines. Businesses in every vertical have already started to leverage the power of IoT platforms to increase their efficiency and performance. This course introduces the students to the three layer architecture of the IoT exploring all the major connectivity options and application layer protocols. It also encourages the students to develop interesting applications using different development boards, sensors and actuators. The course introduces the students to cloud platforms and analytics including security aspects of IoT Application development.

Course Objectives:

- Learn the fundamentals of the Internet of Things.
- Learn the concepts of smart objects & IoT architecture.
- Compare different application and network layer protocols for Internet of Things.
- Appreciate the role and importance of Data Analytics and Security in real word problems and know the role of IoT in various verticals.

Course Outcomes:

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

- Recognize the principles of smart objects and the potential for IoT and comprehend IoT architecture from sensors to the cloud, including edge gateways.
- Determine the static and dynamic performance metrics for core sensors and to understand the architecture of smart sensors for IoT system design.
- Learn how to use various wired and wireless technologies to provide connectivity for IoT applications and select a suitable application-layer protocol among the ones used for IoT applications.
- Recognize how analytics and artificial intelligence algorithms are used in the IoT ecosystem along with IoT security and privacy concerns.

Desirable Knowledge: UE22CS151A-Python, UE22CS151B- Problem solving with , UE22CS252B- Computer Networks

Course Content:

Unit 1 : Introduction to IoT & Architecture: Introduction

IoT Traffic Model, IoT Connectivity, IoT Verticals, Use Cases & Applications, IoT Value Chain. Examples of IoT Use Cases &Applications, IoT Project Implementation, IoT Standards. **IoT Architecture:**Introduction, Factors Affecting an IoT Architectural Model, IoT Architectural Model, IoTWF Model, Data Center & Cloud, Computing (Cloud, Fog & Edge).

14 Hours

Unit 2 : IoT Sensors

Introduction, Sensors & Its Performance Metrics, Sensor Selection, Smart Sensors, MEMS, Sensor Fusion, Self Calibration, Sensors of the Future.

14 Hours

Unit 3 : IoT-Protocols: IoT Wired Connectivity

Introduction, Ethernet, Ethernet TSN, Power Line Communications. **Unlicensed-band Wireless Connectivity:** Introduction, Zigbee, BLE, WiFi, LoRaWAN. **Cellular IoT Technologies:** Introduction, GSM-IoT, LTE, Practical Use Cases**. IoT Data Communication Protocols:** Introduction, HTTP, MQTT, CAOP

14 Hours

Unit 4 : IoT – Analytics, Security & Privacy

Introduction, Data Pipeline, AI, Machine Learning, Supervised Learning Technique: Classification: Decision Tree Example. **IoT Cloud-Based Services & Platforms:** Introduction to AWS, Azure, ThingSpeak for IoT. **IoT Security -** Introduction, IoT Threats, IoT Vulnerabilities, IoT Threat Modeling & Risk, IoT Security Regulations, IoT Privacy Concerns & Regulations, IoT Security & Privacy Examples, IoT & Blockchain.

14 hours

Tool & Languages: Python, Embedded-C, Cloud Platforms, Single Board Computers.

Textbook (s):

1: Dian, F. J, "Fundamentals of Internet of Things", Wiley Professional Development (P&T), 2022.

Reference Book(s)

1: David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry,"IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", 1st Edition, Pearson Education (ISBN: 978-9386873743)
2: "Building Enterprise IoT Applications", Chandrasekar Vuppalapati, CRC Press, Taylor & Francis Group, 2020
3: Arshdeep Bahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities press, 2015

UE22CS342AA4: Applied Cryptography (4-0-0-4-4)

Cryptography is the science of securing data by using mathematical concepts. This course will present the fundamentals of cryptography, as well as its applications and issues of how cryptography is used in practice. Students will have opportunities to dwell well into problem solving and hands-on sessions.

Course Objectives:

- To enable to learn the fundamental concepts of cryptography and utilize these techniques in computing systems.
- To discuss about various symmetric encryption techniques.
- To understand the concept of public key cryptography.
- To introduce message authentication and hash function.
- To provide an overview of authentication techniques using cryptography.

Course Outcomes:

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

- Evaluate classical ciphers and different cryptographic primitives.
- Explain the notions of symmetric encryption and sketch their formal security definitions.
- Describe and implement specifics of prominent public-key crypto systems.
- Evaluate the authentication and hash algorithms.
- Discuss authentication applications.

Course Contents:

Unit 1: Classical Ciphers

Introduction to Cryptography, Cryptanalysis and brute-force attack, Basic cryptographic primitives, Classical ciphers: Substitution - Caesar, Playfair and Hill cipher, Transposition cipher - Rail fence, Columnar and Double columnar, Cryptanalysis of classical ciphers, Mathematical background for cryptography: Modulo arithmetic, GCD, Euclidean algorithm, Introduction to probability, Conditional probability, Law of total probability, Shannon's theorem and Perfect secrecy, One-time pad encryption and its limitations.

14 Hours

Unit 2: Symmetric Key Cryptography

Introduction to symmetric key cryptography, Pseudo Random Number Generator (PRNG), The Feistel cipher, Data Encryption Standard (DES), Analysis, Multiple DES, Security and the avalanche effect, Mathematics- I: Galois fields, Polynomials- Advanced Encryption Standard (AES), Transformation functions, Key expansion, Analysis of AES, Block and Stream ciphers, Block cipher modes of operation, Drawback of symmetric key cryptosystem, Need for asymmetric key cryptography.

14 Hours

Unit 3: Public Key Cryptography and Hashing techniques

Introduction, Mathematics- II: Prime number, Primitive root, Prime factorization, The RSA algorithm, Diffie-Hellman key exchange, Elgamal cryptographic system, Elliptic curve cryptosystem, Digital Certificates, Cryptographic hash functions: Introduction, Applications, Collision resistance, Secure Hash Algorithm (SHA), Message Digest algorithm (MD5).

14 Hours

Unit 4: Hashing Techniques and key

Key-Distribution Centre (KDC), Kerberos, Public Key Infrastructure (PKI) and standard (PKCS), Birthday attack, Entity authentication methods: Password, Challenge-Response, Zero knowledge protocols, Authentication using cryptography use cases: Electronic money, factory setup, web application, WhatsApp, Use Case of Asymmetric and Symmetric Encryption: HTTPS, Metaverse, advancement of cryptography, connect to quantum and blockchain, Introduction to Post-Quantum cryptography

14 Hours

Tools / Languages: SEED labs, Python Programming Language

Textbook (s):

1: Understanding Cryptography: A Textbook for Students and Practitioners, Christoff Paar and Jan Pelzl, Springer 2010.
2: Cryptography and Network Security: Principles and Practice, William Stallings, 7th Edition, Pearson, 2017.
3. Cryptography and Network Security, Behrouz A. Forouzan, 3rd Edition, Tata McGraw Hill, 2017.

Reference Book (s):

1: Introduction to Modern Cryptography", Jonathan Katz, Yehuda Lindell, 2nd Edition, CRC Press, 2015.

UE22CS342AA5: Augmented and Virtual Reality (4-0-0-4-4)

This course presents an introduction to virtual and augmented reality technologies, with an emphasis on designing and developing interactive virtual and augmented reality experiences using blender and Unity3D.

Course Objectives:

- Introduce the use of geometric transformations on graphics objects, their application in composite form and its implementation.
- Impart the basics of computer graphics and Introduce graphics programming using OpenGL and Graphics Pipeline.
- Introduce Virtual and Augmented Reality essentials and understanding human physiological aspects with respect to virtual reality applications and user interfaces.
- Understanding of use of Artificial Intelligence in the field of Virtual and Augmented Reality and applying the concepts to create creative virtual experiences.

Course Outcomes:

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

- Apply techniques and methods of augmenting virtual objects in real space.
- Demonstrate the fundamentals of computer graphics and display pipeline systems and use OpenGL for complex 3D graphical visualization and demonstrate its applications.
- Apply techniques and tool to design a immersive virtual reality experience and graphics in greater depth to more complex aspects of Image Processing, Tracking and Human Computer Interfaces, etc.
- Apply AI models and methods for creating dynamic interactive and adaptable virtual spaces.

Desirable Knowledge : UE22CS252A- Data Structures and its Applications.

Course Content

Unit 1: Geometric Objects and Transformations

Scalars, Points and Vectors, Three-Dimensional Primitives, Coordinate Systems and Frames, Modelling a Coloured Cube, Overview of 2D Transformations: Rotation, Translation and Scaling, Affine transformations, Transformation in Homogeneous Coordinates, Concatenation of Transformations, OpenGL Transformation Matrices, Interfaces to Three Dimensional Applications, Quaternion's.

14 Hours

Unit 2: Graphical System and Programming and 3D Modelling.

The Programmer's Interface, Graphics Architectures, Programmable Pipelines. Graphics Programming: Programming Two Dimensional Applications. The OpenGL: The OpenGL API, Primitives and Attributes, Colour, Viewing, Control Functions, Polygons, Viewing, Control Functions, the gasket Program, Polygon and Recursion, The Three-dimensional gasket, adding interaction, adding menus.

14 Hours

Unit 3: Augmented and Virtual Reality.

Introduction to Augmented Reality: Definition and Scope, A Brief History, Examples, Requirements and Characteristics: Methods of Augmentation, Spatial Display Models, Visual Display, Stationary Tracking Systems, Mobile Sensors. Introduction: What is Virtual Reality, Modern VR Experience. Bird's Eye View: hardware, Software. Eye movement and its implications or VR. Tracking: 2D and 3D orientation, Tracking Position and Orientation, Tracking Attached bodies.

14 Hours

Unit 4: IO modalities, AI, and Behaviour in VR

Computer Vision and Augmented Reality; marker tracking, Multiple-Camera Infrared Tracking, Natural Feature Tracking by Detection, Incremental Tracking, Simultaneous Localization and Mapping, Outdoor Tracking, Interaction: Output and input modalities, Haptic interaction and Multimodal interaction. 3D Scanning of environments. Reactive AI: Adaptability, Complexity and Universality, Feasibility, More Intelligence in the System: Deliberative AI, Reinforcement learning through interaction, Imitation Learning through human demonstration.

14 Hours

Tools/ Languages: C/ C++/ JAVA/ Python using OpenGL.

Text Book(s):

1: “Interactive Computer Graphics - A top-down approach with shader-based OpenGL”, Edward Angel and Dave Shreiner, Pearson Education, Sixth edition, 2012.
2. Steven M. LaValle. Virtual Reality. Cambridge University Press, 2017, http://vr.cs.uiuc.edu/ (Links to an external site.) (Available online for free)
3. Creating Augmented and Virtual Realities, by Erin Pangilinan, Steve Lukas, Vasanth Mohan, PUBLISHED BY:O'Reilly Media, Inc.PUBLICATION DATE:March 2019

Reference Book(s)

1: “Interactive Computer Graphics: A Top -Down Approach with WebGL”, Edward Angel, Pearson Education, 7th Edition, 2015.
2: Unity Game Development in 24 Hours , Geig, Mike. Sams Teach Yourself . Pearson Education, 2014.
3. “OpenGL Programming Guide”: Mason Woo, Jackie Neider, Tom Davis, Dave Shrenier: 3rd Edition, openGl version 1.2, Addision Wesley, 1999.
4. Blender 3D Basic, Gordon Fisher, PACKT Publishing, 2nd Edition (Note: For working with recent version the course material for UNIT 4 can be substituted with appropriate web content.
5. D. Schmalstieg and T. Höllerer. Augmented Reality: Principles and Practice. Addison-Wesley, Boston, 2016, ISBN-13 978-0-32-188357-5

UE22CS342AA6: Human Computer Interaction (4-0-0-4-4)

This course provides an overview and introduction to the field of human-computer interaction, with an emphasis on what HCI methods and HCI-trained specialists can bring to design and development teams for all kinds of products. The course will introduce students to tools and techniques for creating or improving user interfaces.

Course Objectives:

- Get familiar with the psychology underlying user-interface and usability design guidelines keeping in mind human behavioural and perceptual capabilities and limitations that affect interface design.
- Get familiar with the basic principles of goal-directed user interface design and standard patterns and key modeling concepts involved in visual interface design for software interfaces.
- Aware of development methodologies and life cycle models for building user interfaces and prototyping in user interface design and how to test them.
- Get insights with the impact of usable interfaces in the acceptance and performance utilization of information systems.

Course Outcomes:

- Develop and incorporate a Conceptual Vocabulary for analysing Human Interaction with software affordance, conceptual model and feedback.
- Interpret and Exemplify how User-Centered design complements other software process models. Apply low fidelity (lo-fi) prototyping techniques to gather and report user responses.
- Analyze and Define a User-Centred design process that explicitly takes into account the fact that the user is not from the Ecosystem of the Developer.
- Evaluate and Pick appropriate methods to support the development of a specific UI.

Course Content:

Unit 1 : Basics of Design

The Human: I/O Channels, Memory, Reasoning and Problem Solving, The Computer, Devices, Memory, Processing and Networks; **Interaction:** Models, Frameworks, Ergonomics, Styles, Elements, Interactivity, Paradigms, Design Focus, Interaction Design basics.

14 Hours

Unit 2 : Design Thinking

Process, Scenarios, Navigation, Screen Design, Iteration and Prototyping, HCI in software process, Software Life Cycle, Usability Engineering, Prototyping in Practice, Design Rationale, Design Rules, Principles, Standards, Guidelines, Universal Design, Cognitive model.

14 Hours

Unit 3 : Design Analyzing and Models

Socio-Organizational Issues and Stakeholder Requirements, Communication and Collaboration models, GUI Design and Aesthetics, Task Analysis, Task Decomposition, Knowledge based Analysis Technique, Dialog notations and design.

14 Hours

Unit 4 : Notation and Multimedia Design

Petri nets and State Charts, Models of the system, Modeling rich interaction, Case study - A chosen Mobile App as an Interactive System, Groupware, Ubiquitous Computing, Virtual and Augmented Reality, Hypertext, Multimedia and World Wide Web, Augmented Reality Toolkit practice session.

14 Hours

Tools/ Languages: Figma, Adobe Creative Cloud.

Text Book(s):

1: “Human Computer Interaction”, Dix A., Finlay J, Abowd G. D. and Beale R., 3rd Edition, Pearson Education, 2005.

Reference Book(s):

1: “Designing the User Interface”, B. Shneiderman; Addison Wesley 2000 (Indian Reprint).
2: “About Face: The Essentials of Interaction Design by Alan Cooper”, Robert Reimann, David Cronin. Christopher Nooessel, 4th Edition, WILEY, 2014.
3: “Don’t Make Me Think, Revisited: A Common Sense Approach to Web and Mobile Usability”, Steve Krug (Author), 3rd Edition.(e-Book)
4: “The Design of Everyday Things”, Don Norman, 2nd Edition, 2013.

UE22CS343AB1 : Enterprise Business Systems for Engineers (4-0-0-4-4)

The course in Enterprise Business Systems (EBS) offers students a comprehensive understanding of integrated business management systems and their role in streamlining organizational processes. Through theoretical insights, application examples, and case studies, students will understand the functionalities, implementation methodologies, challenges and strategic implications of ERP systems in modern enterprises including associated systems like Supply Chain Management [SCM] and Customer Relationship Management [CRM].

Course Objectives:

The objective(s) of this course is to,

- To learn the strategic importance of Enterprise Resource Planning, Supply Chain and CRM systems in the industry.
- To learn the basics of ERP, the modules of ERP.
- To learn the key selection criteria and implementation issues & risks of ERP
- To be aware of ERP related technologies and ERP Trends.

Course Outcomes:

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

- Identify typical functionality of ERP, SCM & CRM sub-systems.
- Apply criteria to select ERP Package & Consulting Partner.
- Systematically develop plans for an ERP Implementation project.
- Identify critical success factors and associated risks of an ERP Implementation.

Course Content:

Unit 1: Introduction to Enterprise Resource Planning Systems

Introduction, Value Chain Framework, Problems with Disintegrated Data in an Organization, Evolution of ERP Systems, Role of ERP Systems in an Organization, Are ERP Systems Different from Traditional Information Systems , Scope of ERP Systems, General Model of Business and Role of ERP, Major ERP Players, Implementations in India. **Life Cycle of an ERP Implementation Project:** Life Cycle of an ERP Project, ERP Project Teams, Implementation Methodologies, Deployment Methods. **Benefits and Cost of an ERP System:** Benefits, Cost of an ERP Implementation, Cost-Benefit Analysis.

14 hours

Unit 2:Change Management, Re-Engineering and BPM: Change Management: People Issues: Are You Ready for ERP, Factors that Influence Pre-implementation Attitude, How to Enhance Attitude?, Change Management Strategies to Handle Organizational Issues, Creating a Change Management Strategy to Handle Organizational Issues, Tools for Assessing the Organizational Changes. **Re-Engineering:** Processes and their Characteristics, Life Cycle of a BPR Project, Life Cycle of an IT-driven BPR Project, Re-engineering Examples, Case Studies. **Business Process Modelling and Business Modelling**: BPM Introduction, Business Process Hierarchy, Standards for Business Processes and Modelling, Process Modelling Maturity and Multi-Dimensional Modelling, Process Modelling Software, Business Modelling, Integrated Data Modelling.

14 hours

Unit 3: ERP Functional Modules: Human Capital Management: HCM Systems, Leading HR Solutions from ERP Vendors, Strategic Vs. Operational HR Processes and HR Outsourcing, Employee Health and Safety. **Financial Management:** ERP Financial Application, Financial Modules in Detail. **Procurement and Inventory Management:** Procurement, Inventory Management. **Production Planning and Execution:** Understanding MRP II Concepts, How ERP PP module supports MRP II Processes, Critical Master Data Elements, Managing different Production Scenarios.

14 hours

Unit 4: ERP Selection and Emerging Trends: ERP Selection Team, ERP Solution and Vendor Selection, Information Gathering, Preliminary Filtering, Selection Parameters, Request for Proposal (RFP), Gap Analysis, AHP for ERP Selection. **Managing an ERP Project:** Critical Success Factors, Project Risks, Performance Metrics (using Balanced Scorecard). **ERP and Related Technologies:** Electronic Data Interchange (EDI), Supply Chain Management (SCM), Customer Relationship Management (CRM), Product Life Cycle Management (PLM), Data Warehousing and Business Intelligence (DW-BI). **Introduction to Commercial ERP Related Software**: Indian ERP Market, SAP, Oracle, MS Dynamics, SalesForce CRM. Implementations in India; Articles and Case Studies on ERP Trends like Analytics, AI, Cloud and Other Technology Integration.

14 hours

Tools / Languages: Oracle, SalesForce, Web Resources.

Text Book(s):

1. “Enterprise Resource Planning- Text & Cases”, Rajesh Ray, Tata McGraw Hill, New Delhi, 2011.

Reference Book(s):

1. “Enterprise Resource Planning: A Managerial Perspective”, Veena Bansal, Pearson Education India, 2013.
2. “ERP Demystified”, Alexis Leon, McGraw Hill Education, 4th Ed, 2019.
3. "Modern ERP: Select, Implement, and Use Today's Advanced Business Systems" by Marianne Bradford; Pearson, 4th Edition, 2020.
4. “Enhancing Enterprise Intelligence: Leveraging ERP, CRM, SCM, PLM, BPM, and BI", Vivek Kale, Auerbach Publications, 2016.
5. “Supply Chain Management: Strategy, Planning and Operation”, Sunil Chopra, Pearson Education, 7th Edition, 2019.

UE22CS343AB2 : Big Data (4-0-0-4-4)

The course introduces various Big Data technologies that are used to analyze large amounts of data either in batch mode or streaming mode. It focuses on both processing and storage technologies and looks at how algorithms need to be modified to work with large amounts of data. This course requires the student to have a desirable knowledge of Data Structures and its Applications and Design and Analysis of Algorithm.

Course Objectives:

- Provide an introduction to Big Data technologies.
- Introduce computational and storage technologies for Big Data Ecosystem.
- Introduce algorithms for processing Big Data and programming tools focussing on practical issues in working with Big Data.
- Learn application of Big Data techniques to various real-life problems.

Course Outcomes:

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

- Explore various characteristics of Big Data Problems.
- Evaluate principles and design alternative computational/storage technologies for Big Data.
- Design Big Data applications using available infrastructure for Big Data through practical assignments.
- Applying Big Data techniques in real life problems through a group-based project.

Course Content:

Unit 1 : Introduction

Big Data definition, Challenges and opportunities with Big Data, Data intensive scientific discovery and the role of Big Data, History, Map Reduce – Storage (HDFS), Computation model, Map Reduce architecture, Overview of Hadoop Ecosystem, YARN introduction, Case Study: HIVE, Google File System.

14 Hours

Unit 2 : Big data infrastructures (Compute/Storage)

Introduction to sample Big Data Algorithms – matrix multiplication and pagerank, Issues with Hadoop, Relational operators on Map-reduce, Complexity of Big Data algorithms – Communication Cost complexity model. case study: HBase , Cassandra.

14 Hours

Unit 3 : In memory computation

Spark and Scala/PySpark programming model, Transformations and Actions, Spark SQL, Spark architecture – RDD, DataFrames, Wide and Narrow dependencies, Streaming Algorithms –sampling, set membership – Bloom Filters counting, counting unique elements – Flajolet Martin Algorithm.

14 Hours

Unit 4 : Streaming analysis and advanced analytics on Big Data

Streaming analytics use cases, Streaming Spark, Kafka – use cases, architecture, Apache Flink, Clustering algorithms - k means and Collaborative filtering, Scaling Neural Networks for Big Data, Case Study MLLib.

14 Hours

Tools/ Languages: Hadoop, HDFS, Spark, Streaming Spark, HIVE, HBase, Mllib.

Text Books:

1: “Big Data Analytics”, Rajkamal, Preeti Saxena, 1st Edition, McGraw Hill Education, 2019.
2: “Big Data Simplified”, Sourabh Mukherjee, Amit Kumar Das, Sayan Goswami, 1stEdition, Pearson, 2019.

Reference Book(s):

1: “Mining of Massive Datasets”,Anand Rajaraman,Jure Leskovec,Jeffrey D. Ullman, Cambridge Press, 2014.
2: “Big Data Analytics Beyond Hadoop: Real-Time Applications with Storm, Spark, and More Hadoop Alternatives”,Vijay Srinivasa Agneeswaran, Pearson Education, 2014.

UE22CS343AB3: Graphs Theory and its Applications (4-0-0-4-4)

This course focuses on mathematical structure to model the relations between the objects. It also discusses about the basics of graph theory together with a wide range of applications to different branches of Science and Technology, and to real-world problems. The course includes principles of Graph database and social networks.

Course Objectives:

- Familiarize with concepts and abstraction of Graph Theory.
- Up skill students with a computer representation of graphs and algorithms.
- Introduce students with advanced concepts in graph theory and applications. & Graph Database.
- Up skill students with principles and tools for social network analysis.

Course Outcomes:

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

- Understand the graph theory concepts, abstractions and results to model real-world problems.
- Implement high-performance computer representation and graph algorithms.
- Understand various applications of graph theory in varied discipline.
- Apply principles of graph to analyze social networks.

Desirable Knowledge: UE22CS151B – Problem Solving with C, UE22CS252A – Data Structures and its Applications.

Course Content

Unit 1: Introduction, paths, cuts and planar Graphs

Introduction – Review of Representation and Traversals, Walks, Paths, Circuits Euler graphs Hamiltonian paths and circuits – Directed graphs, Digraphs and binary relations, Trees – Properties of trees, Rooted and binary trees, spanning trees, cut sets – Properties of cut set – All cut sets – Fundamental circuits and cut sets – Connectivity and Separability – Network flows, Isomorphism.

14 Hours

Unit 2: Graph Coloring, Graph Applications

Planar graphs –Different representation of a planar graph, Chromatic number – Chromatic partitioning – Chromatic polynomial, Matching, Covering, Four Colour problem, Register Allocation using graph coloring.Minimum Spanning Trees, Shortest Path Problem, Articulation

Points, Network Flow Algorithms, Chinese Postman Problem, Time Table Problem, Strongly Connected Components

14 Hours

Unit 3: Graph Databases and Graphs in Social Networks

Graph Database: RDBMS vs Graph Database, Neo4j Basics, Neo4j Graph Data Model, Nodes, Properties, Relationships, Labels, Neo4j CQL Commands and Functions. Introduction to Graphs in Social Networks:Graphs in Social Networks: Centrality measures: Degree, Betweenness, Closeness, Eigen vector, Katz and PageRank Centrality, Measures of cohesiveness: Degree distribution, Diameters, Transitivity and Reciprocity, Clustering coefficient, Groups and substructures: top-down view – weak and strong component, in-component, out-component, giant component, Bottom-up view: clique, N-clique, N-core.

14 Hours

Unit 4: Graphs in Social Networks and social graph dynamics

Community detection: cluster vs. community, Overlapped Communities by CPM, Girvan Newman algorithm, Louvain algorithm **,** **Generative Models -** Small World: Milgram‟s small world experiment. Scale-free networksRandom Graph Models, Barabasi and Albert's model, Watts- Strogatz small world model, Myopic Search.**Dynamics on Graph** : Herd behaviour, Diffusion- Diffusion of innovation, Bass model of diffusion – epidemic modelling, Cascades.

14 Hours

Tools / Languages: Python, Neo4j, NetworkX

Text Book(s):

1: “Graph Theory: With Application to Engineering and Computer Science”, Narsingh Deo, Prentice Hall of India, 2017.

2: “Graph Databases: New Opportunities for connected data”, 2ndEdition, Ian Robinson, Jim Webber, Emil Eifrem, Oreilly Publications, 2015.

3: “Networks, Crowds, and Markets: Reasoning about a Highly Connected World”, David Easley, Jon Kleinberg, Cambridge University Press, 2010

4: “Social Media Mining”, Reza Zafarani, Mohammad Ali Abbasi, Huan Liu, Cambridge University Press, 2014

5: “Network Science”, Albert-Laszlo Barabasi, e-book

Reference Book(s):

1: “Graph Theory”, F. HARARY, Addison-Wesley,1969.

2: “Discrete and Combinatorial Mathematics”, Ralph P.Grimaldi&B.V.Ramana ,5th Edition, PHI/Pearson education.

3: “Graph Theory with Applications”, J A Bondy and U. S. R Murthy, Elsevier Science Publishing Co. ., Inc.

4: “Graph Representation Learning”, William L Hamilton, Morgan and Claypool Publishers, 2020.

UE22CS343AB4: Bio-Inspired Computing (4-0-0-4-4)

The field of natural computing has been the focus of a substantial research effort in recent decades. These bio inspired computing algorithms have proven to be successful problem solvers across domains as varied as management science, telecommunications, business analytics, bioinformatics, finance, marketing, engineering, architecture and design, to name but a few. This course provides a comprehensive introduction to bio inspired computing algorithms. This course requires the student to have a desirable knowledge of Design and Analysis of Algorithms.

Course Objectives:

- Introduce fundamental topics in bio-inspired computing.
- Build up their proficiency in the application of various algorithms in real-world problems.
- Provide an understanding of a range of features from the biological world that have influenced the world of computing.
- Foster a basic understanding of the nature biological inspiration for AI and Computing - the goals and motivations.

Course Outcomes:

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

- Understand some of the essential features of biologically inspired systems.
- Develop an understanding of simple computer modelling of biological systems.
- Develop a foundation for biological learning models and self-organisation.
- Understand the strengths, weaknesses and appropriateness of nature-inspired algorithms.

Desirable Knowledge: UE22CS241B – Design and Analysis of Algorithms

Course Contents:

Unit 1 : Introduction Natural Computing Algorithms

An Overview, Evolutionary Computing: Introduction to Evolutionary Computing, Evolutionary Algorithms: Genetic Algorithm: Canonical Genetic Algorithm, Design Choices in Implementing a GA, Choosing a Representation, Initialising the Population, Measuring Fitness, Generating Diversity, choosing Parameter Values Extending the Genetic Algorithm: Dynamic Environments, **Structured Population GAs**, Constrained Optimisation, Multi objective Optimisation, Memetic Algorithms, Linkage Learning, Estimation of Distribution Algorithms.

14 Hours

Unit 2 : Evolution Strategies and Evolutionary Programming :

The Canonical ES Algorithm, Evolutionary Programming Differential Evolution: Canonical Differential Evolution Algorithm, Extending the Canonical DE Algorithm, Discrete DE Genetic Programming: Genetic Programming, Bloat in GP, More Complex GP Architectures, GP Variants, Semantics and GP.

14 Hours

Unit 3 : Social Computing:

Particle Swarm Algorithms, Other Foraging Algorithms Search, Particle Swarm Optimisation Algorithm, Comparing PSO and Evolutionary Algorithms, Maintaining Diversity in PSO, Hybrid PSO Algorithms, Discrete PSO, Evolving a PSO Algorithm. Ant Algorithms: A Taxonomy of Ant Algorithms, Ant Foraging Behaviours, Ant Algorithms for Discrete Optimisation, Ant Algorithms for Continuous Optimisation, Multiple Ant Colonies, Hybrid Ant Foraging Algorithms, Ant-Inspired Clustering Algorithms, Classification with Ant Algorithms, Evolving an Ant Algorithm. Honeybee Dance Language, Honeybee Foraging, Designing a Honeybee Foraging Optimisation Algorithm, Bee Nest Site Selection, Honeybee Mating Optimisation Algorithm; Non-uniform Oscillators and Firefly, the model and optimization

14 Hours

Unit 4: Other Social Algorithms and Immune computing:

Artificial Immune Systems Other Social Algorithms: Glow Worm Algorithm, Bat Algorithm, Fish School Algorithm, Locusts. Artificial Immune Systems: The Natural Immune System, Artificial Immune Algorithms, Negative Selection Algorithm, Dendritic Cell Algorithm, Clonal Expansion and Selection Inspired Algorithms, Immune Programming, The Future of Natural Computing Algorithms, Looking Ahead, Open Issues.

14 Hours

Tools/Languages: Matlab, Inspyred pythonic tool

Text Book(s):

1: “Natural Computing Algorithms”, Anthony Brabazon, Michael O’Neill, Seán McGarraghy, Springer, Natural Computing Series,2015.

Reference Book(s):

1:“Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications”,Nunes de Castro, Leandro , Chapman & Hall/ CRC, Taylor and Francis Group, 2007

2: “Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies”,Floreano D. and Mattiussi C., MIT Press, Cambridge, MA, 2008.

UE22CS343AB5 : Advanced Computer Networks (4-0-0-4-4)

Advanced Computer Networks course is designed for students who have basic knowledge of networking (OSI model, TCP/IP, ARP etc.). The course builds its contents by considering a mixed approach of involving theory and practical. From this perspective, the course introduces advanced topics that are basic blocks of networking and covers fundamentals that are used to develop few of the advanced technologies, including routing protocols and switching. The course aims to provide information so that at the end of the course, the students find themselves comfortable in taking either of the direction- industrial job or further research in networking.

Course Objectives:

- Provide in-depth study of dynamic and distance vector routing protocols
- To learn about Layer 2 Switching in detail
- To get an insight of Link State routing protocols
- To explore the new approach of Networking, SDN and NFV

Course Outcomes:

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

- Designing a network topology by demonstrating the ability to configure routers with different routing techniques
- Understand the concepts of Layer 2 switching, VLANs and its usage
- Designing networks with one or multiple routing protocols and investigating its behaviour/features
- Comprehend features of Software Defined Networking (SDN) for next generation systems

Desirible Knowledge : UE22CS252B – Computer Networks.

Course Content:

Unit 1: Dynamic and Distance Vector Routing, Access Lists

IP Routing Process, Static routing – IPv4/IPv6, Floating Static Routes, Load Sharing, Fast and Process Switching Distance Vector Routing – RIPv2 packet format, classless Routing, compatibility with RIPv1, Metric Calculation, Route Summarization, Standard IP and Extended IP access lists, Calling access lists, Named access lists.

14 Hours

Unit 2: Exterior Gateway Routing Protocol (EIGRP), Layer 2 Switching, Spanning Tree Protocol (STP), Virtual LAN (VLAN)

EIGRP – Neighbor discovery, Reliable Transport Protocol (RTP), Diffusing Update Algorithm (DUAL), VLSM and Summary Routes, Metrics – Maximum Paths & Hop Count, Load Balancing with EIGRP, Discontiguous Networks, Auto summarization, passive interface Layer 2 switching, Bridging vs LAN Switching, Forwarding Table, Spanning Tree Protocol and its operations, VLAN Basics, Static and Dynamic VLANs, VLAN Trunking Protocol (VTP), Routing between VLANs, configure VLANs and Inter-VLAN routing, MAC VLANs, VXLAN.

14 Hours

Unit 3: Link State Routing Protocols – OSPF & BGP

Open Shortest Path First (OSPF)v2 – OSPF packet format, Network Types, Designated Routers and Backup Designated Routers (DR & BDRs), OSPF Neighbors, Areas – Intra, Inter, External, Router Types, Virtual Links, Link State Database, LSA Types, Metric Calculation, Summary Routes, Route redistribution Border Gateway Protocol (BGP) – BGP Packet Format, External and Internal BGP

14 Hours

Unit 4: Introduction to Software Defined Networks (SDN) and Network Function Virtualization (NFV)

Segment Routing, SDN Control and Data Plane, SDN Controller and Network-control Applications, OpenFlow Protocol, Data and Control Plane Interaction Network Virtualization architecture, Management, Control and Data Planes, Virtual Switches, Micro-segmentation.

14 Hours

Tools / Languages: GNS3, Cisco Packet Tracer.

Text Book(s):

1. "Routing TCP/IP volume 1", 2nd Edition, Jeff Doyle, Cisco Press 2005.
2. "Computer Networking: A Top-Down Approach", James F. Kurose, Keith W. Ross, 7th Edition, Pearson Publication, 2017 [For SDN]

Reference Book(s):

1: "Software-Defined Networks: A Systems Approach", Peterson, Cascone, O'Connor, Vachuska, and Davie, https://sdn.systemsapproach.org/index.html [For NFV]

2: “TCP IP Protocol Suite”, Behrouz Forouzan, 4th Edition, McGraw-Hill, 2010 [For BGP]

Labs/Assignments:

1. Applying IP addressing scheme to design a network
2. Setting up Dynamic Static Routing for the designed network and observing its behaviour
3. Setting up using different routing protocols. Configuring using EIGRP/OSPF/BGP, observing the metrics, route redistribution
4. For the setup in 3, apply VLAN configuration and access lists and observe its behaviour.

UE22CS343AB6: Computer Network Security (4-0-0-4-4)

This course will give an overview and conceptual understanding of network related security aspects. Students will have opportunities to dwell well into technical "how to" with hands-on sessions and case study discussions.

Course Objectives:

- To provide an overall view of Network Security and introduce the concept of packet analysis.
- To understand the security problems in the design and implementation of the TCP, IP/ICMP, ARP protocols.
- To learn the vulnerabilities in DNS protocol and to implement and experiment with Firewall rules. To provide an overview of network management techniques and implementation of VPN.
- To understand concepts of risk management and security aspects of wireless networks.

Course Outcomes:

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

- Sniff packets from clients and analyse them to extract important info such as headers, passwords etc.
- Launch DoS and MITM attacks using various protocol vulnerabilities and mitigate them.
- Configure firewalls on Linux machines and exploit vulnerabilities on DNS protocol. Design and implement VPN for a secure connection over the internet.
- Master in wireless network security systems in depth and perform effective network management.

Desirable Knowledge: UE22CS252B – Computer Networks

Course Contents:

Unit 1: Introduction, Packet Sniffing & Spoofing, MAC Layer Attacks, Network Layer Attacks
Introduction
CIA principles, Attack surface and types, Assets, Vulnerabilities and Threats, Countermeasures, Privacy, General Data Protection Regulation, Security vs Privacy, Data Breaches. Real Life Examples of Cyber Crime, Security framework, Job outlook. **Packet Sniffing and Spoofing:** Introduction, Sending packets: Network Interface Card (NIC), BSD packet filter (BPF). Packet sniffing: Receiving packets using sockets, Packet sniffing using Raw sockets, Packet sniffing using PCAP API, Processing captured packets. Packet spoofing: Sending normal packets
using sockets, Constructing spoofed raw ICMP packets and UDP packets. Sniffing and then spoofing, Python vs Scapy, Endianness. **MAC layer and attacks:** The MAC layer, ARP protocol, ARP cache poisoning attacks, MITM
using ARP cache poisoning, Demo, Countermeasure. **Network layer:** IP, ICMP and attacks: IP protocol, IP fragmentation, Attacks using IP fragmentation: Problem and solution, Routing, and spoofing prevention, ICMP protocol, ICMP redirect attack, Smurf and other ICMP attacks.

14 Hours

Unit 2: TCP Attacks and DNS Attacks
TCP Attacks : Attacks on the TCP protocols: TCP overview, Send and receive buffers, SYN flood attack: TCP 3-way handshake, the SYN flooding attack, Launching the attack using Netwox and C, Countermeasure. TCP reset attack: TCP reset attack on Telnet, SSH and video streaming connections. TCP session hijacking attack: TCP session and session hijacking, Launching the attack, Hijacked TCP connection. Reverse shell: working, redirecting IO to TCP connection, Creating reverse shell **DNS Attacks:** DNS hierarchy, zones and servers, DNS query process, Constructing DNS request and response using Scapy, DNS attacks: Overview, Local DNS cache poisoning attack.

14 Hours

Unit 3: Remote DNS Attack & Countermeasure, Firewall, IDS, IPS, Honeypots
Remote DNS Attack : Remote DNS cache poisoning attack (Kaminsky attack), Reply forgery attacks from malicious DNS servers, Countermeasure against DNS spoofing attacks, DoS attacks on DNS servers. Case Study – 1. **Firewall:** Introduction, Requirements of a firewall, Firewall characteristics and Access policy, Types of firewalls, NG firewall, Shortcomings, Firewall location and configuration: DMZ networks, Firewall topologies. Introduction, Build a simple firewall, Netfilter, iptables firewall in Linux, Stateful firewall and connection tracking, Application/Proxy firewall and Web proxy, Evading firewalls. **Intrusion Detection and Prevention:** Intruders, Intrusion detection, Analysis approaches, Host-based intrusion detection, Network-based intrusion detection, Distributed or hybrid intrusion detection, Honeypots, Example system: Snort.

14 Hours

Unit 4: Virtual Private Networks, Wireless Security, SIEM & SOC
Virtual Private Network: Introduction, Why VPN, analogy, and tunneling. Overview of TLS/SSL VPN: Establishing a tunnel, Forwarding, and releasing IP packets, TLS/SSL VPN details. Building, Setup and Testing VPN. Bypassing Firewall using VPN. Case Study – 2.

The Heartbleed Bug and Attack: Introduction and the Heartbeat protocol, Launching the attack, Fixing the Heartbleed bug. **Wireless Security:** Communications and 802.11 WLAN standards: Wired Equivalent Privacy (WEP), Wireless Protected Access (WPA), IEEE 802.1x, 802.11i/ WPA2, Wireless Network Threats. **SOC and SIEM.**

14 Hours

Note: Hands-on experience for relevant topics in the form of Lab and/or Assignment are given. Relevant cyber security case study for undergraduate students is discussed.

Tools / Languages: SEED Ubuntu VM, Wireshark, Snort, Netwox, Scapy.

Text Book(s) :

1: "Internet Security: A Hands-on Approach", Wenliang Du, 2nd Edition, 2019.
2: "Internet Security: A Hands-on Approach", Wenliang Du, 3rd Edition, 2022.
3: "Computer Security: Principles and Practice", William Stallings and Lawrie Brown, 3rd edition, 2014.

Labs

#	Lab Task
1	Packet Sniffing & Spoofing using Scapy
2	Packet Sniffing & Spoofing using PCAP
3	ARP Cache Poisoning Attack Lab
4	ICMP Attack Lab
5	TCP Attack Lab
6	Local DNS Cache Poisoning Attack
7	Remote DNS Cache Poisoning Attack
8	Firewall Lab
9	Firewall Evasion Lab
10	VPN lab
11	Heartbleed Attack Lab (Optional)

Labs

#	Assignment Task
1	Understand and earn Plagiarism Certificate - Indiana University
2	Case Study I : iPremier
3	Case Study II : University of Virginia

UE22CS343AB7: Robotics with ROS (4-0-0-4-4)

This course introduces students to fundamental skills for effective robot programming. Covering robotic design, sensor and actuator functionality, and the Robot Operating System (ROS), the course emphasizes practical application.Through hands-on projects, students learn to select components and master ROS, including integration with Arduino and Raspberry Pi. By course end, students can design and implement robotic applications, leveraging their expertise in sensor and actuator selection, ROS programming, and hardware integration.

Course Objectives:

- To introduce students the fundamentals of robotic programming and summarize the different types of robot sensorsand actuators.
- To introduce students to the criteria for selecting a sensor and actuator for a particular application.
- To understand the Robot Operating System (ROS) fundamentals and introduce students to the criteria for selectinga sensor and actuator for a particular ROS robotic application.
- To understand integration of Arduino and Raspberry Pi boards with ROS and familiarize with various hardwarebased robotic application.

Course Outcomes:

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

- Understand robotics design, implementation and gain the knowledge on fundamentals of robotic programming.
- Comprehend, classify and analyse the behaviour of different types of sensors and actuators.
- Understand the ROS fundamentals.
- Design robotic applications using ROS and integration of Arduino and Raspberry Pi boards with ROS.

Course Content:

Unit 1: Introduction to Robotics and ROS

Robot Introduction- Defining a Robot, Robot Controllers-Major Components, Robot Vocabularies- Robotics Middleware Basics. Robot Operating System (ROS):ROS Basics-ROS Equation, History of ROS, Sensors and Robots Supporting ROS, ROS Architecture and Concepts, ROS Filesystem Level, ROS Computation Graph Level, ROS Community Level.

14 Hours

Unit 2: Programming the Robot's Sensors

A close look at Sensors, Programming the Robot's Sensors, Programming the Actuators, Building Robot's Softbot. ROS Fundamentals: Ubuntu Linux for Robotics-Ubuntu Graphical User Interface, Shell Commands, C++ and Python for Robotic Programming- Basic Concepts with Examples

14 Hours

Unit 3: ROS Programming

Creating ROS Workspace and Package, Using ROS Client Libraries, Programming Embedded Board using ROS-Interfacing Arduino with ROS, ROS on a Raspberry Pi. Robotic Projects using ROS:Introduction to Wheeled Robots, Building Robot Hardware-Block Diagram and Assembling Robot Hardware, Programming Robot Firmware

14 Hours

Unit 4: ROS Navigation

Localizing the robot in a map, ROS Navigation Stack-hardware requirement-navigation packages, path planning, motion planning of robot – software requirement and configuration. Navigation using Reinforcement learning and Imitation learning.

14 Hours

Tools: C++, Python, Java, ROS, Rviz, Gazebo and Webots.

Recommended Materials

- Robot Operating System (ROS): http://wiki.ros.org/ROS/Tutorials
- ROS2 tutorials: https://docs.ros.org/en/foxy/Tutorials.html

Text Book(s):

1: Lentin Joseph, Robot Operating System (ROS) for Absolute Beginners: Robotics Programming MadeEasy, 1st Edition, APress, 2018.

2. Jonathan Cacace; Lentin Joseph, Mastering ROS for Robotics Programming: Design, build, and simulate complexrobots using the Robot Operating System, 2nd Edition, Packt Publishing, 2018.

Reference Book(s):

1. Hughes, C. and Hughes, T., Robot programming: a guide to controlling autonomous robots. Que Publishing, 2016
2. Quigley, M., Gerkey, B. and Smart, W.D., Programming Robots with ROS: a practical introduction to the RobotOperating System. " O'Reilly Media, Inc.", 2015
3. Anil Mahtani, Luis Sanchez, Enrique Fernandez, Aaron Martinez, Lentin Joseph. ROS Programming: BuildingPowerful Robots. Packt Publishing, 2018

UE22CS351B: Cloud Computing (4-0-2-5-5)

The cloud computing course introduces not only the various technologies that go into building a cloud native application, but also how cloud systems are designed. The student is introduced to various tools and design techniques/tradeoffs. It also gives a flavour for the business relevance/ethics of using cloud computing. This course requires the student to have a desirable knowledge of Computer Networks and Operating System.

Course Objectives:

- Introduce the rationale behind the cloud computing revolution and cloud native application architecture
- Explore the concepts of Virtualization and Containerization.
- Explore the concepts of Distributed storage and its various techniques
- Design distributed systems for scalability and expose the student to various tradeoffs in designing cloud architectures.

Course Outcomes:

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

- Comprehend the technical and business rationale behind cloud computing. Decide the model of cloud computing to use for solving a particular problem.
- Implement Microservice architecture through Containers and Orchestration tools. Analyze virtual machines and containers.
- Experiment with cloud storage Models such as object stores, key value stores.
- Apply the critical constraints such as Performance, scalability and security to the designed distributed system.

Desirable Knowledge: UE22CS252B- Computer Networks, UE22CS241B- Operating Systems.

Course Content:

Unit 1 : Cloud Programming Models

Parallel computing, Grid computing, Introduction to Cloud Programming Models and service Models, Introduction to technology challenges with Distributed & Cloud computing, Business Drivers - deployment models, Cloud architecture and IaaS programming model, Web Services and REST, PaaS Programming Model, Communication using Message queues- Pub Sub model, SaaS Programming model – Microservices and differences with the traditional monolithic model; challenges of migrating monolithic applications**.**

14 Hours

Unit 2 : Virtualization

Hypervisor - Types, Para virtualization and Transparent virtualization, Software - Trap and Emulate virtualization, Software - Binary translation, Goldberg Popek principles for Virtualization, Hardware - AMDv/Intel, Memory - Shadow page tables, Memory - Nested page tables, IO, VM Migration, Lightweight Virtualization - Containers and Namespaces, Deployment of cloud native applications through Docker – Unionfs, DevOps, Orchestration and Kubernetes.

14 Hours

Unit 3 : Distributed Storage

Types of Cloud storage - Block, Object stores, Replication, lag, multileader replication, Leaderless replication, Partitioning - key-value data, Consistent hashing, Partitioning - rebalancing partitions, Request routing, Consistency Models, CAP Theorem, Transactions, Two-phase commit.

14 Hours

Unit 4 : Cloud Controller, Performance, Scalability and Security

Master-slave v/s p2p models, Resource allocation, Scheduling algorithms, Cluster coordination – consensus, Fault Tolerance - faults and partial failures, Failure detection - checkpointing and application recovery, Unreliable communication, Cluster coordination - leader election, distributed locking, Case Study: Zookeeper/Raft - distributed consensus infrastructure.

Scaling computation - reverse proxies, Scaling computation - hybrid cloud and cloud bursting, Multitenancy, Multitenant databases, Cloud security requirements - physical/virtual security, security design patterns,

Authentication in the cloud: Keystone/IAM, Cloud Threats – DoS, Economic Denial of Sustainability**.**

14 Hours

Tools/Languages: Amazon AWS (or equivalent), AWS Skill Builder, AWS Educate, Qwiklabs, Docker, Kubernetes, Jenkins, Zookeeper, Github, NoSQL database, Flask, Python, GoLang.

Lab/ Hands-on:

1: Migrating a monolithic e-commerce application to a microservices architecture
2: Building a Task Management Application with Raft Consensus Algorithm and MySQL.
3:Microservices communication using RabbitMQ
4: Building a Distributed Key-Value Store with etcd
5: Back Up service using docker and Kubernetes
6: Building an E-commerce Microservices Application on Cloud using Docker, Kubernetes, Jenkins, and Git

Text Book(s):

1: "Distributed and Cloud Computing", Kai Hwang, Jack Dongarra, Geoffrey Fox.ISBN: 978-0-12-385880-1, Morgan Kaufmann, 2012.
2:"Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems", Martin Kleppmann. O'Reilly, 2017.

Reference Book(s):

1: "Docker in Action", Jeff Nickoloff, Manning Publications, 2016.
2: "Cloud Native DevOps with Kubernetes", John Arundel and Justin Domingus, OReilly, 2019.
3: "Moving to the clouds: Developing Apps in the new world of cloud computing", Dinkar Sitaram and Geetha Manjunath. Syngress, 2011.

UE22CS352B: Object Oriented Analysis and Design (4-0-2-5-5)

In this course students will learn to perform Analysis on a given domain and come up with an Object-Oriented Design (OOD). Various techniques will be discussed and practiced which are commonly used in analysis and design phases in the software industry. Unified Modelling Language (UML) will be used as a tool to demonstrate the analysis and design ideas and object-oriented programming language such as Java/C++ would be used to implement the design. The theory is supplemented with implementations which are demonstrated/ practiced in class which provides the hands-on experiences of implementing the patterns.

Course Objectives:

- Familiarize students with static and dynamic models of object-oriented analysis and modelling using the unified modelling language (UML)
- Introduce students to object oriented programming concepts using object oriented programming language
- Make students appreciate the importance of system architecture design in software development
- Introduce the students to understand the importance of GRASP and SOLID design principles along with Design patterns

Course Outcomes:

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

- Construct static models, use cases and class models, followed by analysing the dynamics of the system using activity, sequence, state and process models. Depict the architecture of a software system by using component and deployment models
- Use the concepts of classes and objects of object-oriented programming to model a complex system
- Use GRASP and SOLID principles in the design of software application and apply Creational software design patterns for variety of application scenarios
- Apply Structural and Behavioural software design patterns for variety of application scenarios. Understand Anti-pattern.

Course Content:

Unit 1: Object Oriented Design and UML Diagrams

Requirements, Modelling and Analysis: Introduction to UML, Use Case Modelling: Use Cases Diagrams. Domain models, Class Modelling: UML Class Diagrams, OO relationships, CRC Diagrams, Component model, Deployment model, Activity Modelling: UML Activity Diagrams and Modelling, Guidelines. Behaviour Modelling: Sequence Diagram, UML State Machine Diagrams and Models, Advanced State Models.

16 Hours

Unit 2: Object Orientated Programming and Architecture design

Overview of Object Orientated Programming using Java/C++ –Classes, Inheritance, Overloading, Overriding, Abstract classes, Interfaces. OO Development process, System Design and Frameworks, Architectural patterns: MVC, Layered Pattern, Client-Server Pattern, Event-Driven Pattern, Microkernel Pattern, Microservices Pattern.

12 Hours

Unit 3: Design Principles & Patterns

GRASP and its application to Object Design, Creator, Information Expert, Low Coupling, Controller, High Cohesion, Polymorphism, Pure Fabrication, Indirection and Protected Variations SOLID: Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion OO Design Principles and Sample Implementation of Patterns. Introduction to Design Patterns, selection and usage of a design pattern, Creational Design Patterns Theory and Implementation: Singleton, Factory, Builder and Prototype

14 Hours

Unit 4: Design Patterns

Structural Patterns – Adapter, Façade, Proxy and Flyweight, Behavioural Patterns – Chain of Responsibilities, Command, Interpreter, Iterator. Anti-patterns – Introduction and classification, Project Management, Architecture and Development anti-patterns (1 anti-patterns of each type)

14 Hours

Tools / Languages: Star UML, Object Oriented Programming Language (Java/C++)

Text Book(s):

1: "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development", by Craig Larman, 3rd Edition, Pearson 2015.
2: "Software Architecture Patterns" by Mark Richards, 2nd Edition, OReilly 2022

Reference Book(s):

1: "Object-Oriented Modelling and Design with UML", Michael R Blaha and James R Rumbaugh, 2nd Edition, Pearson 2007.

2: “Design Patterns: Elements of Reusable Object-Oriented Software” by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, 1st Edition, Pearson 2015.
3: “Fundamentals of Software Architecture: An Engineering Approach” by Mark Richards & Neal Ford, First Edition, 2020.

Hands-on/Assignment/Laboratory/Project

- Lab assignment on Use case diagram.
- Lab assignment on Class diagram.
- Lab assignment on Activity and State diagrams.
- Lab assignment on Object Oriented Programming Concepts.
- Lab assignment on Architectural Patterns
- Self Learning Assignment on Serialization and Multithreading
- Hands-on Assignment on MVC Framework.
- Assignment on Design Patterns.
- Mini Project using any Architectural pattern incorporating all learning of the course.
- Experiential learning – Java/C++ Certification

UE22CS341B: Compiler Design (4-0-0-4-4)

Language design and implementation is an active topic in programming, and will likely always be. How we program and the tools we use, changes constantly. We try new ideas and come up with better or alternative approaches frequently. Any language that doesn‟t continue to adapt will fall into disuse, and any tool chain that remains stagnant will be forgotten. Hence knowledge of compilers in order to tweak these changes in the language design is a must for a Computer Science Engineer. This course requires the student to have a desirable knowledge of Data Structures and its Applications and Automata Formal Languages and Logic.

Course Objectives:

- Introduce the major concept areas of language translation and compiler design.
- Develop a greater understanding of the issues involved in programming language design and implementation.
- Provide practical programming skills necessary for constructing a compiler. Develop an awareness of the function and complexity of modern compilers.
- Provide an understanding on the importance and techniques of optimizing a code from compiler's perspective.

Course Outcomes:

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

- Use the knowledge of patterns, tokens and regex for solving the problems in the field of data mining
- Analyze and design the semantic behaviour of a compiler. Choose the appropriate compiler internal representation for different kinds of compiler tasks.
- Translate a source-level language into a low-level compiler internal representation.
- Optimize the performance of a program in terms of speed and space using new code optimization techniques.

Desirable Knowledge: UE22CS252A- Data Structures and its Application, UE22CS243A- Automata Formal Languages & Logic.

Course Content:

Unit 1: Compilers: Introduction, Lexical Analysis, Top-down Parsers

The Language Processing System, The Phases of a Compiler, The Grouping of Phases into passes.

Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, Design of a Lexical Analyzer Generator. The role of the parser, Syntax Error Handling, Error-Recovery Strategies. Top-down parsing: Recursive Descent Parser (RDP) with Backtracking, LL(1) Parser.

14 Hours

Unit 2: Syntax Analysis: Bottom-up Parsers, Syntax-Directed Translation

Bottom-up parsing: Shift-Reduce Parsing, LR (0), SLR, viable prefixes, CLR, LALR. Syntax- directed definitions, Evaluation orders for SDD‟s: S-attributed SDD, L-attributed SDD.

14 Hours

Unit 3: Implementation of Syntax-Directed Translation Schemes and Intermediate Code Generation

Applications of Syntax-Directed Translation - SDD for Syntax Trees, Expressions, Basic Types and flow control statements. Syntax-

directed Translation Schemes – Parser Stack Implementation of Postfix SDT's, SDT's with actions inside Productions, SDT's for L-Attributed Definitions. Implementing L-Attributed SDD‟s: Bottom-Up Parsing. Variants of Syntax Trees – Directed Acyclic Graphs for Expressions, Three-Address Code – Addresses and Instructions, Quadruples, Triples, Indirect Triples, SSA Form, Control Flow Graph.

14 Hours

Unit 4: Machine Independent Code Optimization, Code Generation and Run Time Environment

Machine Independent Optimization: Different Optimizations, Optimization of Basic Blocks. Data Flow Analysis: Live-variable analysis, Next-use algorithm. Storage Organization, Different Allocation Strategies, Stack Allocation of space, Access to Non local Data on the stack. Code Generation: Issues in the design of a code generator, the target language, addresses in the target code, static allocation, stack allocation, run-time addresses for names. A Simple Code generator - The Code generation algorithm.

14 Hours

Tools/Languages: Lex/flex and YACC/Bison.

Text Book(s):

1: “Compilers–Principles, Techniques and Tools”, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffery D. Ullman, 2nd Edition, Pearson Education, 2009.

Reference Book(s):

1. “Modern Compiler Design”, Dick Grune, Kees van Reeuwijk, Henri E. Bal,Ceriel J.H. Jacobs, Koen Langendoen, 2nd Edition, 2012.
2. AW Appel, J Palsberg, Modern Compiler Implementation in Java, Cambridge University Press, 2002.
3. AW Appel, M Ginsburg, Modern Compiler Implementation in C, Cambridge University Press, 2004.

UE22CS342BA1: Supply Chain Management for Engineers (4-0-0-4-4)

This course explores supply chain management (SCM) processes. It aims for students to develop the skills to understand, design and evaluate techniques in the realm of supply chain management. It covers fundamentals relevant to SCM, current SCM challenges and trends in the industry including planning and prediction, techniques and tools using AI technologies.

Course Objectives:
- Understand the fundamentals of supply chain management.
- Learn the various SCM stages and strategies of planning, sourcing, inventory, distribution and logistics.
- Explore case studies of SCM applications to understand various industrial scenarios.
- Explore current trends and AI techniques applied to solve real-world SCM challenges.

Course Outcomes:
At the end of the course, the student will be able to:
- Identify various stages of SCM including Logistics.
- Evaluate KPIs and select suitable strategies for optimized supply chain performance.
- Evaluate and select the strategies for planning and managing Inventories and Logistics.
- Apply ML techniques to predict trends, optimize inventory, and improve logistics.

Desirable Knowledge: Enterprise Business Systems

Course Content:

Unit 1: Introduction to SCM

Overview of supply chain management, Building a strategic framework to analyze Supply Chains. Understand supply chain scope, strategic fit, its drivers and performance metrics. Managing Material and Information Flows. Intro to supply chain networks and their design. Various strategies of planning, sourcing, inventory, distribution and logistics related to SCM – Opportunities, Competitive advantage, Real-time Supply Chains. Value Stream Mapping.

14 Hours

Unit 2: Procurement and Supply Chain Planning (Demand & Supply)

Procurement processes, Supplier selection and performance assessments. Intro to Demand Forecasting models, predictive analytics for supply chain planning. Techniques for aggregate planning of demand and supply in supply chains, managing demand variability and accuracy. Industry trends in procurement – E-procurement, Case studies. **14 Hours**

Unit 3: Planning and Managing Inventory and Logistics Management in SCM

Introduction to planning and management of Inventory in an enterprise. Techniques for warehouse management and logistics management for distribution. Logistics stakeholders and modern logistics concepts, Outsourcing logistics, Logistics KPIs. Intro to Reverse Logistics, and Lean Logistics. Enhancing transportation and logistics with real-time tracking and monitoring using IoT and AI-driven route planning and last-mile delivery. Case studies in inventory and logistics management. **14 Hours**

Unit 4: Drivers, Analytics and Current Trends in Supply Chain Management

Understanding and managing cross functional drivers in supply chain. Decision making in supply chains - sourcing, pricing, risk management and compliance. Basics of AI, Blockchain, IoT concepts applied to SCM. Current trends in SCM: Green SCM, Lean and Global SCM, SCM Analytics. Exploration of AI techniques, tools and technologies for SCM. Awareness of current challenges and research areas of SCM in the industry.

14 Hours

Tools / Languages: SCM applications and tools, ML tools, Case studies, Web resources.

Textbook:

1. Supply Chain Management: Strategy, Planning and Operation, Sunil Chopra, Pearson, 7th Revised Edition, 2024.

Reference(s):

1. Essentials of Supply Chain Management, Michael H Hugos, John Wiley & Sons, 3rd Edition, 2018.
2. Principles of Supply Chain Management: A Balanced Approach, Joel D Wisner, Cengage, 5th Ed, 2019.
3. Strategic Supply Chain Management: The Five Core Disciplines for Top Performances, Cohen Shoshanah & Joseph Roussel, Mc-Graw Hill, 2nd Ed, 2013.
4. Logistics and Supply Chain Management, Martin Christopher, Pearson, 2022.
5. Supply Chain Logistics Management, Donald Bowersox, et al, Mc-Graw Hill Publishing, 5th Edition, 2024.
6. Supply Chain Management: A Logistic Approach, John J Coyle, Cengage, 10th Ed, 2019.
7. A Roadmap to Green Supply Chains: Using Supply Chain Archaeology and Big Data Analytics, Kevin L Lyons, Industrial Press, 2015.
8. Lean Supply Chain and Logistics Management, Paul Myerson, Mc-Graw Hill, 2012.
9. Supply Chain Analytics, T. A. S. Vijayaraghavan, Wiley, 2021.
10. Logistics & Supply Chain Management - Course (nptel.ac.in).
11. Blockchain, IoT, and AI Technologies for Supply Chain Management, CRC Press, 2023.
12. Web resources.

UE22CS342BA2 - Algorithms for Information Retrieval (4-0-0-4-4)

This course covers the basic and advanced algorithms and techniques for Information retrieval and web applications. This course focuses on Index building, document ranking, use of machine learning in Information retrieval , recommendation algorithms and design of intelligent web applications. This course requires the student to have a desirable knowledge of Design and Analysis of Algorithms.

Course Objectives

- Understand the architecture, models, and algorithms used in Information Retrieval.
- Understand the basic principles and implementation of Indexing and Search.
- Understand the use of machine learning in Information Retrieval and Web Applications.
- Understand multimedia and multimodal information retrieval.

Course Outcomes:

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

1. Implement an efficient index for document collection.
2. Perform searches on a document collection, rank, and evaluate results.
3. Apply suitable techniques for multimodal information retrieval
4. Apply Machine Learning techniques in Information Retrieval Systems a Web Applications.

Course Contents:

Unit-1: Basics of Information Retrieval

Introduction to Information Retrieval, Background, Architecture and Strategies of Information Retrieval (IR) Systems,IR Models, Boolean and Extended Boolean Models, Dictionary, Vocabulary, Positional Postings, Phrase Queries and Tolerant Retrieval, Indexing and Vector Space Model, Evaluation of IR Algorithms for Indexing and Index Compression, Vector Space Model for Scoring, tf-idf and Variants.

14 Hours

Unit-2: Ranking and Web Search Basics

Efficient Scoring and Ranking, Parametric and Zone Indexes, Tiered Indexes, Query Term Proximity, Query Parser, Aggregating Scores, Performance Measurement, Web Applications and Search Algorithms, Relevance Feedback, Query Expansion, Other IR Models, Web Search Basics

14 Hours

Unit-3: Link Analysis, Multimodal Information Retrieval

Economic Model of Web Search, Improving Search Results, Link Analysis, The Page Rank Algorithm, Other Search Algorithms, Scalability Issues in Search. Search User Experience, Web Crawling and Indices, Link Analysis, Building a Complete Search System, Lucene as a Search Engine, and Other search engines like Solr, Everything, and Google. Multimodal Information Retrieval content-based visual information retrieval: metadata searching, Query by Example, Semantic Retrieval, Machine Learning Approaches, content comparison using image distance measures. Multimedia and Multimodal Information Retrieval: Introduction, Requirements, Applications, challenges, Architecture, Metadata, Techniques for Content Processing.

14 Hours

Unit-4: Question Answering, neural models for IR

Question Answering, neural models for IR Question Answering, Neural models for IR QA as an Information Retrieval task, Factoid QA models, Entity Linking models, Knowledge-based QA, and Pretrained models for QA.Recent and trending research papers in QA.

14 Hours

Textbook(s):

1: “Introduction to Information Retrieval”, Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze, ISBN: 9781107666399, Cambridge University Press, 2009.
2. “Search Computing”, Challenges and Directions: chapter 8: Multimedia and Multimodal Information Retrieval.

3. “Speech and Language Processing”, Third Edition, Daniel Jurafsky, James H. Martin, Chapter 14: Question Answering and Information Retrieval, 2023.

Reference Book(s):

1: “Algorithms of the Intelligent Web”, Haralambos Marmanis, Dmitry Babenko, Manning Publishers, 2011.

UE22CS342BA3 : Image Processing and Computer Vision (4-0-0-4-4)

igital Image Processing deals with processing images that are digital in nature. Improving the quality of images for human perception and understanding, extracting useful information for decision making and efficient storage are some of the driving factors behind image processing techniques/algorithms. The course on Digital Image Processing introduces the learner to various image processing techniques, algorithms and their applications.

Course Objectives:

- Understand the principles underlying imaging and assess the quality of an image based on its visual content, histogram and statistical measures
- Gain an insight to image transforms and enhancement techniques in the frequency domains
- Be able to apply morphological operations and segmentation techniques on monochrome and color images
- Learn different feature extraction techniques for classification

Course Outcomes:

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

- Assess the quality of an image and identify the appropriate processing technique that needs to be applied in the spatial domain
- Apply image enhancement techniques frequency (Fourier) domain and interpret images in the transform domain
- Design and evaluate methodologies for segmentation of region of interest from binary, gray scale or color images
- Use appropriate feature extraction techniques for tasks such as classification and be able to work with data-driven approaches (deep learning architectures) for computer vision applications

Desirable Knowledge: UE22CS241B – Design and Analysis of Algorithms.

Course Content:

Unit 1: Introduction and Enhancement in the Spatial Domain
What is Digital Image Processing, examples of fields that use DIP, Fundamental Steps in Digital Image Processing, elements of visual Perception, Basic Concepts in Sampling and Quantization, Representing Digital Images, Spatial and Gray-level Resolution, Zooming and Shrinking Digital Images, Some Basic Relationships Between Pixels, Linear and Nonlinear Operations. Image Enhancement in the Spatial Domain: Some Basic Gray-level Transformations, Histogram Processing, Enhancement Using Arithmetic/Logic Operations, Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters, and Combining Spatial Enhancement Methods.

14 Hours

Unit 2: Image Enhancement in the Frequency Domain, Transforms and Compression
Image Enhancement in the Frequency Domain: Introduction to the Fourier Transform and the Frequency Domain, Smoothing Frequency-Domain Filters, Sharpening Frequency Domain Filters, Homomorphic filtering. Image transforms (such as, DCT, Harr, DWT) for applications. Image Compression: Fundamentals - Image Compression Models, Some encoding techniques (including, block processing and transform-based techniques).

14 Hours

Unit 3: Morphological Image Processing, Preliminaries of segmentation and Color Image Processing
Dilation and Erosion, Opening and Closing, the Hit-or-Miss Transformation, Some Basic Morphological Algorithms. Image Segmentation: Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding, Region-Based Segmentation. Color Models, Pseudocolor Image Processing, Basics of Full-

Color Image Processing, Color Transformations, Smoothing and Sharpening, Color Segmentation, Noise in Color Images.

14 Hours

Unit 4: Feature extraction, pattern classification

Boundary Preprocessing, Boundary Feature Descriptors, Region Feature Descriptors, Principal Component Analysis (use of Eigen images), Whole image features, Scale Image Feature Transform, Image pattern classification using minimum distance classifier and prototype matching, deep convolutional neural networks and some of its recent variations for computer vision applications and state-of-art techniques and applications in computer vision.

14 Hours

Tools / Languages: Matlab, Python Programming Language.

Text Book(s):

1: "Digital Image Processing", Rafael C Gonzalez and Richard E. Woods, Prentice Hall, 4th Edition, 2018.

Reference Book(s):

1: "Digital Image Processing and Analysis", Scott E. Umbaugh, CRC Press, 2014.
2: "Digital Image Processing", S. Jayaraman, S. Esakkirajan, T. Veerakumar, McGraw Hill Ed. (India) Pvt. Ltd., 2013. 3: "Digital Signal and Image Processing", John Wiley, 2003.
4: "Computer Vision A Modern Approach", D. A Forsyth and J. Ponce, Pearson Education, 2003
5:"Computer Vision: Algorithms and Applications", Richard Szeliski, Springer, 2ndEdition, 2022.

UE22CS342BA4: Natural Language Processing (4-0-0-4-4)

The goal of this course is to focus on processing of text data as found in natural language usage. The key problem discussed in this course is that of understanding the meaning of text by various types of learning models including the recent approaches using deep learning and the significance of the NLP pipeline in that meaning disambiguation process. The course also discusses disambiguation of syntax as a step of meaning disambiguation process. This course requires the student to have a desirable knowledge of Machine Intelligence.

Course Objectives:

- Learn the phases of NLP and the building of Language Models.
- Learn how lexical and distributional semantics can be used for semantic disambiguation in NLP.
- Focus and learn various learning models related to sequence labelling and syntactic disambiguation in NLP.
- Introduce the deep learning techniques and its applications in Natural Language Processing

Course Outcomes:

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

- Have a very clear understanding of the phases of NLP and the building of Language Models.
- Analyze and apply comfortably appropriate branch of semantics depending on the problem being solved.
- Understand various sequence labelling approaches, its applications and syntactic ambiguity removal in NLP.
- Design and implement neural language model, NLP applications using neural techniques and utilize various transfer learning approaches in NLP.

Desirable Knowledge: UE22CS352A-Machine Learning

Course Content:

Unit 1: Introduction to NLP

Introduction, Types of ambiguity in natural language processing, Text normalization, Morphological parsing of words – Porter stemmer, Lemmatization and Stemming, Sentence segmentation. Noisy Channel model: Real world spelling error, Minimum edit distance algorithm, Concept of noisy Channel Model. Language Model: N-grams, ngram language model, smoothing, discounting and back-off, Kneser-Ney smoothing, interpolation, perplexity as an evaluation measure.

14 Hours

Unit 2: Lexical and Vector Semantics

Word senses and relations between word senses, WordNet: A Database of Lexical Relations; Word sense disambiguation, Semantic relatedness, Lexicons for sentiment and affect extraction: available sentiment and emotion lexicons. Vector Semantics and Embeddings: Words and vectors, TF IDF, Pointwise Mutual Information, Measuring similarity, Using syntax to define a word's context, Evaluating vector models, Dense vectors via SVD Distributional Hypothesis, Neural Embedding: skip gram and CBOW Pre-trained word representations: Word2Vec, Improving Word2vec, Limitation of distributional methods.

Self-Learning component: FastText, Glove.

14 Hours

Unit 3: Handling sequences of text and Parsing - Disambiguating Structure

Sequence labelling: Hidden Markov Model, POS Tagging example, Discriminative Sequence labelling with features-Conditional Random Field. Other sequence labelling applications – Named Entities and Named Entity Tagging, POS Tagging using discriminative models i.e. Maximum Entropy Markov Model (MEMM)

Self-Learning component: Sequence labelling using RNNs and LSTMs

Constituency parsing: Ambiguity presented by parse trees, CKY parsing, CCG Parsing, Partial parsing – chunking. Statistical Parsing : Probabilistic Context Free Grammar, Probabilistic CKY parsing of PCFG, Problems with PCFG, Probabilistic Lexicalized CFG Introduction to dependency parsing: Dependency relations, Dependency Formalisms, Dependency Tree Banks. Evaluating parsers.

Self-Learning component: Span-based Neural Constituency Parsing

14 Hours

Unit 4: Coreference resolution, Transformers and Pretrained Language Models

Coreference resolution: Forms of referring expression, algorithms for coreference resolution – mention pair and mention ranking model, mention detection, classifiers using hand-built features.

Self-Attention Networks: Transformers, Transformers as Language Models, Sampling, Pretraining Large Language models. (Top notch/trending LLM model to be discussed during the course delivery), LLM's Vs SLM's.
Fine-Tuning and Masked Language Models: Bidirectional Transformer Encoders (BERT), Training Bidirectional Encoders, RoBERTa. In Context Learning, Instruction Fine Tuning, Prompt Engineering.

14 Hours

Tools / Languages: Tensorflow, Scikit Learn, Python 3.x. CoreNLP, Natural Language Toolkit (NLTK), TextBlob,Gensim, SpaCy,PyTorch-NLP , OpenNLP, Hugging Face, OpenAI API.

Text Book:
1: "Speech and Natural Language Processing", Daniel Jurafsky and James H. Martin, 3rd edition online, Feb 3 2024. The more up to date 3rd edition draft is available at http://web.stanford.edu/~jurafsky/slp3/. (text book revision is continuous process according to author)

Reference Book:
1: "Introduction to Natural Language Processing", Jacob Eisenstein, MIT Press, Adaptive computation and Machine Learning series, 18th October, 2019. The open source softcopy is available at githubhttps://github.com/jacobeisenstein/gt-nlpclass/blob/master/notes/eisenstein-nlp-notes.pdf.

UE22CS342BA5 : Blockchain (4-0-0-4-4)

Blockchain having wide impact and potential growth for change around the world. It is changing how business is executed. It is important to understand why Blockchain is different and how it works in comparison with technologies of the past. This course requires the student to have a desirable knowledge of Data Structures and its Applications.

Course Objectives:

- Learn a conceptual view of Blockchain for the new applications that they enable.
- Apply the Blockchain for various applications to provide a secure way of data access using cryptographic functions.
- Learn various consensus mechanisms to implement for various real time applications.
- Familiarize with the Blockchain deployment tools.

Course Outcomes:

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

- Analyze how the traditional databases can be replaced with Blockchain for the real time applications.
- Integrate various cryptographic algorithms in to Blockchain.
- Apply various consensus mechanisms to the real world Blockchain applications.
- Evaluate the setting where a Blockchain based structure may be applied, its potential and its limitation.

Desirable Knowledge: UE22CS252A-Data Structures and its Applications.

Course Contents:

Unit 1: Introduction and Data flow and Integrity
Key Blockchain Concepts, Nodes, Cryptocurrency, tokens, Public Ledger, Peer to peer Network, Types of Blockchain, Permissioned Blockchain model, Permission-less Blockchain model, Demonstration of Blockchain Construction steps, Demonstration of ether scan and Goerli/Sepolia etherscan. Cryptography- Need, history, features, Private and public keys, Types of cryptography, Digital signatures, Hash functions, SHA-256, Patterns of hashing, Hash Pointer, Markle tree, Ledgers, Transactions and trade, the public witness, Computers that witness.

14 Hours

Unit 2 : The structure of the network: consensus algorithm
Case study: Bitcoin Blockchain Network, Creation of metamask wallets and performing transaction.
Introduction to distributed consensus: What, why, Challenges, Proof of Work, Proof of Stake, Delegated Proof of Stake, Proof of Authority, Proof of Elapsed Time, Proof of Scope, Proof of Space, Proof of Burn, RAFT, PAXOS, Byzantine Fault Tolerance System, PBFT.

14 Hours

Unit 3: Second generation applications of Blockchain technology
Smart contracts: origins and how they function, Creating and deploying smart contracts, Decentralized applications, Dapps construction, Decentralized Autonomous Organizations (DAOs)- Need, Principal agent dilemma, components, The DAO Story, Legality of DAPPs and DAOs.
Solidity- Variable, Functions, modifiers, view, pure, fallback, overloading, in-built mathematical and cryptography functions, Withdrawal pattern, Restricted Access

14 Hours

Unit 4: Blockchain Security and use cases
Hyperledger Fabric: Blockchain-as-a-service (BaaS), Architecture and core components, Hyperledger fabric model, Creation of a simple DAPP. Blockchain vulnerabilities, Smart contract vulnerabilities, Blockchain on CIA security triad, Blockchain based DNS security platform, deploying blockchain based DDOS protection. Use cases: Public Sector, Finance, Supply Chain. Research Aspects in Blockchain

14 Hours

Tools / Languages: Solidity, Remix, Ganache, Metamask.

Text Book(s):
1: Introduction to Blockchain Technology, Tiana Laurence, 1st edition, Van Haren Publishing, 2019. 2. 2. Blockchain Technology from Theory to Practice, Sudeep Tanwar, 1st edition, Springer, 2022.

Reference Book(s):

1. Hands-On Cyber security with Blockchain: Implement DDoS protection, PKI-based identity, 2FA, and DNS security using Blockchain, Rajneesh Gupta, 1st edition, 2018.

2: Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Narayanan, Bonneau, Felten, Miller and Goldfeder, Princeton University Press, 2016.

UE22CS342BA6: Digital Forensics and Incident Response (4-0-0-4-4)

Digital forensics is the core set of principles and processes necessary to produce usable digital evidence and uncover critical intelligence. Digital Forensics course provides a deep understanding of the techniques to gather, protect and report the digital confirmations

Course Objectives:

- Introduces to the history and evolution of Digital forensics, Principles, challenges and Incident Response Provide an understanding of the various sources of evidence and the procedures for acquiring the evidence
- Learn about, file structures and Windows file forensics and practice with applicable tools.
- Focus on basics of Linux and MAC OS , study on Network Forensics and Browser forensics
- Focus on mobile device forensics ,email investigations , anti-forensics and Learn on the procedure for final report writing as per the court of law

Course Outcomes:

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

- Understand the Principles of Digital Forensics, Evolution of Computer Forensics, Scope of Cyber Forensics Investigation process and Incident response
- Know the various sources of Digital evidence and acquire the evidence by applying the standard acquisition procedures
- Understand the various operating system file systems and able to perform a better forensic analysis of the same and able to Detect network intrusions and unauthorized access,
- Perform basic mobile investigation and e-mail investigations using appropriate tools. Write professional reports for Legal consumption.

Course Content:

Unit 1 : Introduction to Digital Forensics and Incident Response With Essential Technical Concepts

Introduction to forensics science, Basics of Digital forensics, International standards and practices, Role of examiner in judicial system, Digital forensic: goals, categories ,Challenges, investigation types, Forensics readiness, Digital forensics life cycle, Digital forensics Vs other computing domain **Incident Response**: Attack Lifecycle, Incident Response-Introduction, Team, Incident Response Plan (IRP), Lifecycle, Methodology, Goals, Tools **Acquiring-Analyzing Digital Evidence withEssential Technical Concepts** :Digital evidences types, File structures, understanding and working of the hard disks-HDD,SSD, Data recovery considerations and Understanding: Allocated and unallocated space, Data Persistence, Page file(Swap space) Initial Response and First Responder Tasks- Search and Seizure, First responder tool kit and tasks, Documenting digital crime scene, Packing and transporting digital evidences, Conducting interview, Acquiring digital evidence, Analyzing Digital Evidence

14 Hours

Unit 2 : Filesystems - Windows Systems, Windows Registry

Filesystems - Windows file systems- File Systems- understanding and examining FAT, NTFS, File allocation table(FAT), New Technology File System(NTFS), Windows Registry: windows system artifacts, Introduction Deleted Data Hibernation File (Hiberfile.Sys) Registry Print Spooling Recycle Bin Metadata Thumbnail Cache Most Recently Used (MRU) Restore Points and Shadow Copy, Windows forensic analysis- Timeline Analysis, file recovery, Windows registry analysis, Deleted registry key recovery.

14 Hours

Unit 3 : Introduction to Linux and Mac OS X Systems ,Network Forensics and Web browser Forensics

Linux Systems and Artifacts- Linux file systems (Ext2/Ext3), file system layer, filename layer, metadata layer, data unit layer, deleted data, Linux logical volume manager, Linux boot process and services, Linux System Organization and Artifacts, Unix/Linux Forensic Investigation: Unix/Linux forensics, investigation steps and technologies, Principles of file carving **Mac OS X Systems and Artifacts**- OS X File System Artifacts-HFS+ Structures, OS X System Artifacts. **Network Forensics**: Networking Fundamentals and Types of Networks, Network Forensics Overview, Securing A Network, Developing Procedures for Network Forensics Network Security Tools ,Network Attacks, Forensic Footprints, Network Evidence and Investigations, Seizure of Networking Devices,Applying Forensic Science To Networks **Web browser forensics**- Internet overview, IE, Microsoft web browser, Firefox, Google Chrome, web browser investigation tools.

14 Hours

Unit 4: E-Mail forensics, Mobile Device Forensics, Anti-forensics and Report Writing

Email Forensics- Steps in email communications, List of E-mail protocols, E-Mail header examination, email forensics, recovering emails

Mobile device forensics- Cellular Networks and How They Work, Overview of Cell Phone Operating Systems, Potential Evidence Found on Cell Phones, Collecting and Handling Cell Phones as evidence, Cell Phone Forensic Tools, mobile device forensic investigation, storage location, acquisition methods

Anti-forensics: Introduction to Antiforensics, Classification of antiforensics techniques, antiforensics Practices- Data Wiping and Shredding, Trail Obfuscation, Encryption, Data Hiding, stenography techniques and tools, Introduction to malware analysis and malware forensics **Report Writing**: Prep work for report Writing, Structure of the report, Characteristics of a good report, Document design and good writing practices, Legal Acceptance , digital forensics report writing for High-Tech Investigations.

14 Hours

Tools / Languages :Open source tools on Forensics.

Text Book(s):

1: "The basics of digital forensics: the primer for getting started in digital forensics", Sammons, J. Elsevier, 2012.
2: "Digital Forensics Basics: A Practical Guide Using Windows OS", Hassan, N. A. A press 2019.
3: "Practical Cyber Forensics- An Incident-Based Approach to Forensic Investigations", by Niranjan Reddy, A Press, 2019.
4: "Introductory Computer Forensics-A Hands-on practical Approach", by Xiaodong Lin, Springer, 2018.

Reference Book(s):

1: "Guide to Computer Forensics and Investigations", Bill Nelson, Amelia Phillips and Christopher Steuart, Course Technology, Cengage Learning, 2010.
2: "Digital forensics with open source tools", Altheide, C., & Carvey, H. Elsevier, 2011.
3: "Digital Forensics Workbook_-Hands-on Activities in Digital Forensics", Michael K Robinson ,CreateSpace Independent Publishing Platform, 2015.

UE22CS342BA7 - Digital Twin and eXtended Reality (4-0-0-4-4)

This course explores key concepts in Augmented Reality (AR), Virtual Reality (VR), and Digital Twin (DT) technologies. It covers AR/VR fundamentals like 3D transformations, OpenGL, and quaternion-based rotations. Digital Twin essentials include types, modelling approaches (data-driven, physics-driven, hybrid), and case studies across industries like healthcare and transportation. System modelling and simulation principles focus on discrete event systems, performance evaluation, and validation. Finally, the course integrates cybersecurity for Digital Twins, addressing threats, privacy, trust, and resilience through frameworks like blockchain.

Course Objectives:

- Understand the fundamental concepts and technologies of Digital Twin and Extended Reality, including AR and VR, 3D transformation, AI/ML, and data visualization.
- Develop the skills needed to design and implement Digital Twin and Extended Reality solutions, including the ability to create mock, functional, and executable twins.
- To build skill in modelling-based system engineering, focusing on discrete event systems
- Collaborate with peers on practical projects to gain hands-on experience in developing intelligent systems and applying Digital Twin and Extended Reality solutions to real-world problems.

Course Outcomes:

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

- Students will be able to demonstrate a deep understanding of the fundamental concepts and technologies of Digital Twin and Extended Reality, including AR and VR, 3D transformation, AI/ML, and data visualization.
- Students will be able to model discrete systems and apply AR techniques to visualize the simulated system..
- Students will be able to apply Digital Twin and Extended Reality solutions to various industries, including industrial IoT, critical infrastructure, Agri infrastructure, and connected vehicles, and assess the benefits and limitations of these solutions.
- Students will be able to collaborate effectively with peers on practical projects and demonstrate the ability to apply Digital Twin and Extended Reality solutions to real-world problems, such as improving operational efficiency, enhancing safety, and reducing costs.

Desirable Knowledge: UE22CS252A- Data Structures and its Applications.

Course Content:

Unit 1:Introduction to AR and VR

Overview of AR and VR technologies, Introduction to the OpenGL graphics pipeline, Rotation, Translation and Scaling, 3D transformation, homogeneous transformations, First OpenGL program, Controls and Animations, Vector Space, Affine Space and Euclidian Space, Affine Transformation, Transformation in Homogeneous Coordinates, 3D transformation, Homogeneous Transformations, Concatenation of Transformation, Transformation Matrices, Interface to Three dimensional Applications, Quaternions Basics, Quaternions, multiplication and Rotation, Quaternion Visualization

14 Hours

Unit 2: Digital Twin Essentials

The Big Picture of Digital Twins, History of the Digital Twin, Origin of the Digital Twin concept, Digital Twin and Product Life cycle Relationship, Types of Digital Twin: Discrete Vs Composite, Product versus facility, Types of Digital Twin: Simulation versus operational, Analytics versus physics-based, Characteristics of a Digital Twin, Digital Twin Architecture, Industrial Digital Twin applications, Examples of mock, functional, and executable twins, Data Driven Modelling, Physics Driven Modelling, Hybrid (Data-Physics) Modelling, Examples of Physics Driven Modelling: DT of a Propeller of a Drone, DT of a ceiling fan, Bio-Mechanical DT of human body; Examples of Data Driven Modelling: DT for RUL of a Battery, DT for traffic Mobility in Bangalore, DT of RUL of GUT; Hybrid Modelling: Heart Digital Twin, DT for Solar Array, DT for CAR using OBD., Case Study: Manufacturing, Case Study: Healthcare, Case Study: Buildings, Case Study: Transportation & Logistics

14 Hours

Unit 3: System Modelling and Simulation

Introduction to Simulation, System and System Environment, Discrete Event System Simulation, General Principles, The Event Scheduling/Time Advance Algorithm, Input Modelling, Estimation of Absolute Performance, Verification, Calibration and Validation.

14 Hours

Unit 4: Digital Twin & Cyber Security.
Digital twins and cybersecurity, Security Framework, Digital twins threat modelling, Common attacks on digital twins, Common attacks on digital twins, Digital twin authentication and identification challenge, IDS, IPS, Authentication Methods, Communication Channel Protection, building cyber resilience in digital twins, Privacy Framework, Lack of Privacy, and trust, Privacy by Design, Enhancing trust with block chain integration.

14 Hours

Tools/ Languages: C/ C++/ JAVA/ Python using OpenGL.

Textbooks:
"Interactive Computer Graphics - A top-down approach with shader-based OpenGL", Edward Angel and Dave Shreiner, Pearson Education, Sixth edition, 2012.
Building Industrial Digital Twins by Shyam Varan Nath & Pieter van Schalkwyk, by Packt Publishing Ltd.
Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol: Discrete-Event System Simulation, 5th Edition, Pearson Education, 2010.

Reference Books
Diab, W. W., A. Ferraro, B. Klenz, S. W. Lin, E. Liongosari, W. E. Tannous, and B. Zarkout. "Industrial IoT Artificial Intelligence Framework." (2022): 1-59.
El Saddik, Abdulmotaleb, ed. Digital Twin for Healthcare: Design, Challenges, and Solutions. Elsevier, 2022.

Web References:
https://in.mathworks.com/campaigns/offers/next/digital-twins-for-predictive-maintenance/modeling-methods.html
https://learn.microsoft.com/en-us/training/paths/build-mixed-reality-azure-digital-twins-unity/

UE22CS342BA8: Topics in Wireless and 5G Networks (4-0-1-4-4)

Wireless Mobile Networking is a dynamic field that has spurred tremendous excitement and technological advances. This course will cover topics in wireless networking and introducing 5G networks. The objective of the course is to introduce students to recent advances in mobile networking, with an emphasis on practical design aspects of mobile systems.

Course Objectives:

- Introduce the different trends of wireless network technologies
- Applications of various wireless communication fundamentals based on its usage
- Provide cellular systems and standards for different generations, and explore cellular concept of design fundamentals such as Frequency Reuse, Handoff etc.
- Discuss design parameters and applications of computing and architecture for different mobile standards

Course Outcomes:

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

- Develop knowledge on wireless LANs and its evolution
- Understand the concepts of wireless communications and wireless networking
- Learn cellular concepts, e.g. frequency reuse and multiple access technologies
- Understand cellular evolution from 1G to 5G and associated mobility management

Desirible Knowledge : UE22CS252B – Computer Networks.

Unit 1 :Overview of Wireless communication

Introduction, Wireless Local Area Networks: IEEE 802.11, 802.11 Frame Format, Basic Access Methods – CSMA/CA, ALOHA, P-persistent CSMA, Non-persistent CSMA, Distribution Coordinating Function (DCF), DIFS, SIFS, Point Coordination Function (PCF), Light Weight Access Point Protocol (LWAPP), IEEE 802.1x, 802.11a,h,k,p, WPAN Technologies: Bluetooth, Bluetooth Protocol Stack, Piconets, NFC, 6LOWPAN, LPWAN-LORA, Wireless Local Loop (WLL)-Local Multipoint Distribution System (LMDS), MMDS, WiMAX - QDMA, Adhoc Networks and Routing, MANET & VANET, Wireless Sensor Networks, Self-Organizing Networks, Wireless Mesh Networks, RFID: Concept, frequency band, classification of RFID tags, applications

14 Hours

Unit 2 :Wireless Communication Fundamentals

Multiple Access for Wireless Systems – FDMA, TDMA, CDMA, Multiple-Input and Multiple-Output (MIMO), Coding Techniques – Pulse Code Modulation (PCM), Differential PCM (DPCM), Adaptive PCM (ADPCM), Delta Modulation (DM), Wireless Modulation Schemes – Binary Phase-Shift Keying (BPSK), Quadrature Phase-Shift Keying (QPSK), Gaussian Minimum-Shift Keying, Frequency Spectrum, Long Range Communication- Satellite Communication, Smart Antennas, Wireless networking and security issues in 802.11 – Static filtering based on MAC address, Wired Equivalent Privacy (WEP), WPA & WPA2, 802.11e QoS issues – DCF & HCF, Scanning, Increased Bit Error Rate (BER), Multipath Propagation, Authentication, Path Loss, RF signal interference, traffic and resource allocation – flow control, error control, mobility, routing, Channel Allocation Scheme, Power Management

14 Hours

Unit 3 : Fundamentals of Cellular System

Evolution of Wireless Network generations (1G, 2G, etc), Mobile Radio standards – AMPS, N-AMPS, GPRS, GSM, UMTS, CDMA 2000, roaming, Cellular Concept – cell structure, cluster, frequency reuse, basic cellular system: mobile terminal, base station (BS), mobile switching center (MSC), home location register (HLR) and visitor location register (VLR), traffic and control channel (forward and reverse).

Handoff Strategies: Concept of handoff, Types of handoff – hard and soft handoff, Queued Delay, MAHO (Mobile Assisted Handoff), Improper handoff, Umbrella cell approach, Improving coverage and capacity in cellular systems: Cell splitting, Microcell Zone concept, Repeaters for range extension. **(**

14 Hours

Unit 4 : Digital Cellular Mobile Standards

Global System for Mobile Communication (GSM) – features, architecture, GSM call routing, stages of call processing in GSM, Signaling System No. 7 (SS7), Need for 3G, 4G and 5G technologies, UMTS/W-CDMA standard – features, architecture, specifications and other procedures, Next generation mobile standards – features of 4G & 4G LTE, 5G

14 Hours

Text Book(s):
1: “Wireless and Mobile Network Architectures”, Lin Yi-Bang and Clamtac Imrich, John Wiley & Sons 2001

Disclaimer:
There is no fixed text book. Lectures will be drawn from several sources (books, articles, papers) as most available texts do not cover all of the course material. The text book listed covers some of the topics that are relevant to the material presented.

Reference Book(s):
1: “Wireless Communications: Principles and Practice”, Theodore S. Rappaport
2: “Wireless and Mobile Networks: Concepts and Protocols”, Dr. Sunil Kumar S. Manvi, Mahabaleswar S. Kakkasageri, Wiley India
3: “Wireless Communication Networks and System”, by Cory Beard and William Stallings,1st edition, Pearson, 2015.
4: “Wireless Communication and Networks: 3G and Beyond”, ITI Saha Mishra, McGraw-Hill Education
5: “Wireless Network Security: A Beginner’s Guide”; by Tyler Wrightson, McGraw-Hill Education

UE22CS342BA9: Generative AI and its Applications(4-0-0-4-4)

Large Language Models (LLM) in Generative AI represent a significant leap forward in the field of artificial intelligence, leveraging vast amounts of data and advanced algorithms to understand, generate, and interact with human language in a highly sophisticated manner. These models are capable of producing human-like text, understanding context, and even creating content that is coherent and contextually relevant. Their applications are diverse and far-reaching, encompassing areas such as customer service automation, where they can handle inquiries and provide support; content creation, where they generate articles, stories, and marketing copy; and education, where they assist in tutoring and personalised learning experiences and many more applications.

Objectives:

- Gain an in-depth understanding of Large Language Models (LLMs), their architecture, evolution, and applications across different domains.
- Develop hands-on skills in implementing and fine-tuning LLMs for various NLP tasks, using state-of-the-art tools and frameworks.
- Learn and apply advanced techniques in prompt engineering, retrieval-augmented generation (RAG), and multimodal LLMs to solve complex problems.
- Understand and address the ethical implications, potential biases, and privacy concerns associated with the use of LLMs.

Outcomes:

- Demonstrate a solid grasp of LLM architecture, including transformer models like BERT, GPT, ELMo, RoBERTa, and BART, as well as advanced techniques in prompt engineering and RAG.
- Successfully implement and fine-tune LLMs for specific NLP tasks, utilizing frameworks like HuggingFace, and develop multimodal applications that integrate text and image processing.
- Create innovative solutions by applying advanced prompt engineering, reasoning techniques, and agentic workflows, enhancing the capabilities of LLMs in real-world applications.
- Evaluate and mitigate bias, toxicity, and privacy issues in LLM outputs, ensuring responsible and ethical use of AI technologies in various applications.

Unit 1: LLM Architecture and Models

LLM Basics and evolution, NLP:Word Embeddings, POS and NER, Text Classification, LLM Architecture, Transformer Anatomy, BERT, GPT, ELMo, ROBERTa, BART architectures,
Case study:Loading the LLM using file system, client-server pattern,NLP tasks using transformers from HuggingFace, Tokenization and Embedding using HuggingFace Models

14 Hours

Unit 2: Prompt Engineering and RAG

Prompt Engineering techniques ,Zero/few shot inferencing, Chain/Tree/Graph of Thought, Reasoning and Act, LangChain basics, Naive Retrieval Augmented Generation: Chunking, Embedding, Vector stores, processing different data types and formats, MoE, Advanced RAG technique, Building Q&A systems with Naive RAG and Advanced RAG techniques, Bias and toxicity evaluation.

14 Hours

Unit 3: Agent and Multimodal LLM

Agentic Workflows using Microsoft Autogen, CrewAI. GAN, Diffusion networks and Image generation
Multimodal LLMs: CLIP, BLIP models.

14 Hours

Unit 4: Fine Tuning

LLMfinetuning: Data resolution/precision, Quantization, finetuning principles, Prompt tuning and soft prompts, Finetune own LLM, Harm of LLM, Data behind LLM, LLM scaling and privacy.

14 Hours

Tools/Languages: Python, HuggingFace, LM Studio, Kaggle

Text Book:

1. Quick Start Guide to Large Language Models: Strategies and Best Practices for Using ChatGPT and Other LLMs by Sinan Ozdemir, Addison-Wesley Professional, Edition 1 2023

Reference Books:

1. Pretrain Vision and Large Language Models in Python: End-to-end techniques for building and deploying foundation models on AWS,by Emily Webber, packt publishing,Edition 1, 2023
2. Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more by Denis Rothman (Author) , publishing,Edition 1, 2023
3. Understanding Large Language Models: Learning Their Underlying Concepts and Technologies by by Thimira Amaratunga, APress, Edition 1 , 2023
4. Retrieval-Augmented Generation (RAG): Empowering Large Language Models (LLMs) by Dr. Ray Islam (Author) ,Edition 1,2023

UE22CS342BA10: Cloud Security (4-0-0-4-4)

This course focusses on the essentials of Cloud security and offers a practical exposure to securely create or access the cloud services. The course emphasizes on securing data and applications in public cloud environment.

Course Objectives:

• Understand key security challenges in Cloud (i.e., Shared) environment.

• Learn the best practices in Cloud setup and configurations and analyse the security impact of misconfigurations.

• Apply data security controls to protect data in cloud environment and select the appropriate security controls for a given cloud network security architecture

• Learn about securing cloud applications and understand Cloud Security controls, Governance, and applicable industry frameworks, standards

Course Outcomes:

On successful completion, the students will be able to:

- Implement data/service security features, define security groups, enabling inbound and outbound filtering, and identity access management
- Critically review cloud security threats and propose protection services for cloud applications
- Implement logically isolated virtual networking environments
- Evaluate and adapt governance, compliance, operational auditing, and risk auditing cloud technologies.

Desirable Knowledge:

Foundational knowledge of Computer networks Operating system, and data security concepts.

Course Content:

Unit 1: Cloud Security Essentials and IAM in Public Cloud Infrastructure Introduction to Cloud Computing, IaaS, PaaS and SaaS and Service models, Private/Public/ Hybrid and Multi Cloud models, Introduction to Cloud Security, NIST Cyber security Framework, Shared Responsibility Model-cloud Provider and Cloud Customer, Cloud Security Fundamentals-Types of Misconfigurations. IAM Life cycle, Building and Managing IAM in AWS, Access Policy and Authentication using Key vault

14 Hours

Unit 2: Network and Data Security in Public Cloud Infrastructure Zero trust architecture in Single Cloud, zero trust architecture in Hybrid/Multi Cloud infrastructure. AWS Network Security, Amazon VPC security groups, AWS shield, AWS Firewall Manager, AWS Network firewall, Network ACLs, DNS firewall, Data Security- Securing data at rest, Data Security- Securing data at transit, Data in process.

14 Hours

Unit 3: Securing Cloud Applications in Public Cloud Infrastructure Container services, what is Microservice Architecture? Challenges of Microservice Architecture, Service Mesh Architecture, NoSQL injection, Serverless Architecture/Framework, FaaS, CI/CD Pipeline, DevSecOps.

14 Hours

Unit 4: Cloud Security Governance Cloud Controls Matrix (CCM) – cybersecurity control framework for cloud computing, CIS (Center for Internet Security) Critical Security Controls, Cloud logging and Auditing, Overview of CSPM, CWPP Solutions, Security guidance for critical areas of focus in cloud computing by CSA (Cloud Security alliance), Case Study: Exploration of AWS Cloud Security Dashboard.

14 Hours

Hands-on Sessions

Lab 1: Hands on- Exploiting Misconfigurations
Lab 2: IAM in Single Cloud infrastructure
Lab 3: VPC Lab on AWS- Allow traffic on Specific Port/Type
Lab 4: Secure Workloads using DevSecOps

Tools: AWS Security Services

References:

1. Mastering AWS Security by Albert Antony
2. Practical Cloud Security by Chris Dotson Copyright © 2019 Chris Dotson. All rights reserved. Printed in the United States of America. Published by O'Reilly Media
3. https://cisecurity.wistia.com/medias/tcelnntxig
4.https://aws.amazon.com/security/

UE22CS343BB1: Heterogeneous Parallelism (4-0-0-4-4)

This course focuses on parallel heterogeneous architectures as well as programming models and imparts pragmatic
skills to program using parallel programming languages and frameworks trending in the industry.

Course Objectives:

- Familiarize with various parallel heterogeneous architectures, associated techniques and programming models.
- Acquaint with Memory Consistency & Coherence.
- Acquaint with Concurrency Bugs and Resolution Techniques.
- Familiarize with opportunities and challenges in Parallel Programming using popular languages & frameworks.

Course Outcomes:

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

- Understand the underpinnings of Parallel Heterogeneous Architectures and Parallel Computing Techniques.
- Understand Memory Consistency Models and Coherence Techniques, Techniques for Parallelism Bugs' Resolution
- Program using popular Parallel Programming Frameworks and Languages trending in the industry.
- Engineer high performance migration of varied applications appreciating and assimilating varieties of parallelism.

Desirable Knowledge: UE22CS151B – Problem Solving with C, UE22CS251B – Microprocessor and Computer Architecture.

Course Content:

Unit 1: Fine Grained Parallelism

Review on Parallelism and Performance, Instruction Level Parallelism and Enhancement Techniques, Prediction and Speculation, Code Optimization, Cache optimized Programming. Laws of Parallelism, Data, Task and Pipeline Parallelism, Pthreads.

14 Hours

Unit 2: Coarse Grained Parallelism &Parallel Algorithms

Multithreaded and Multi-Core architectures, Parallel Algorithms, Task Decompositions and Mapping. GPUs and GPGPUs, GPU Architectures, Many-Core Heterogeneous Architectures.

14 Hours

Unit 3: Parallelism Frameworks

OpenMP, Race Conditions, Deadlocks & Debugging, Memory Models for Parallel Programming, Memory Consistency Models, Message Passing Programming,

14 Hours

Unit 4: GPU Programming & Hardware Accelerators

CUDA Programming Hardware acceleration platforms (FPGAs and TPUs), Architecture and organization of modern FPGAs, Case Studies, Concurrency in Main stream Languages.

14 Hours

Tools/Languages: pthread, OpenMP, CUDA, openCL.

Text Book(s):

1: "Parallel Programming: for Multicore and Cluster Systems", Thomas Rauber, Gudula Runger, 3 rd Ed., Springer, 2023.

Reference Book(s):

1: "Computer Architecture: A Quantitative Approach: John Hennessy, David Patterson", 6th Edition, Morgan Kaufmann, 2017.

2: “Programming Massively Parallel Processors”, David Kirk and Wen-mei Hwu, 3rd Ed, Morgan Kaufmann, 2016.
3: “Computer Systems: A Programmer's Perspective”, Randal E. Bryant, David R. O'Hallaron, Pearson, 2nd Ed, 2016.4: 4: “Parallel Programming for FPGAs”, Ryan Kastner, Janarbek Matai, and Stephen Neuendorffer, Creative Commons, 2018.

UE22CS343BB2: Topics in Deep Learning (4-0-0-4-4)

Deep Learning has received a lot of attention over the past few years and has been employed successfully by companies like Google, Microsoft, IBM, Facebook, Twitter etc. to solve a wide range of problems in Computer Vision and Natural Language Processing. In this course we will learn about the building blocks used in these Deep Learning based solutions. At the end of this course students would have knowledge of deep architectures used for solving various Vision and NLP tasks. This course requires the student to have a desirable knowledge of Machine Intelligence.

Course Objectives:

- Familirise various CNN architectures and itroduce transform leraning.
- Acquire knowledge on sequence models like RNN, LSTM and GRU.
- Upskill students in generative models and meta learning.
- Attain knowledge on reinforcement learning and other latest deep learning models

Course Outcomes:

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

- Implement a Neural Network using Tensor Flow and Keras
- Classify images using CNN.
- Solve time-series related problems with RNN.
- Generate data in the form of images using GAN.

Desirable Knowledge: UE22CS352A - Machine Learning.

Unit 1:Introduction to Deep Learning

Introduction**,** Activation functions, Loss functions, Batch Normalization, Regularization and Optimization.**Convolutional Neural Network(CNN):** Introduction, Filters, Feature Maps, Max- Pool Layers, Other Pooling Types, Back Propagation. Convolution Architectures - Alexnet, ZFNet, VGGNet, GoogleNet, ResNet. RCNN, FRCNN, Faster RCNN, YOLO V5. **Transfer Learning**: Introduction, Motivation, Variations, TL Architecture of CNNs. Hands-on: Assignment on CNN & TL.

14 Hours

Unit 2:Recurrent Neural Networks (RNN) and Transformers

Introduction-Recurrent Neurons, Memory Cells, Variable-Length Input-Output Sequences, RNN Architecture, Sequence learning problem, BPTT-Back Propagation Through Time, truncated BPTT, Vanishing and Exploding Gradient, Bidirectional RNN, LSTM Cell and GRU Cell, Text Classification with RNN, Encoder/Decoder architecture, Seq2Seq model with Attention , Transformer model and BERT architecture, Transformer Attention and its implementation.

Hands-on: Problem using RNNs, Transformers and TL / predefined model **14 Hours**

Unit 3:Generative Models & Meta Learning: Introduction to Autoencoders , Regularization in autoencoders, Denoising autoencoders, Sparse autoencoders, Contrastive autoencoders , Variational Auto Encoders(VAEs). **Generative Adversarial Networks**(GANs)- Architecture and Training Methods, Image Generation, DCGAN, Style GAN, WGAN, Applications.**Graphical Neural Networks(GNN)**: Introduction to GNNs, Graph Convolution Networks, Applications. **Meta Learning**: Introduction to Meta Learning, MAML, FOMAML, Adaptive Neural inductive Learning, Siamese Networks.

Hands-on: Problem on Fashion **MNIST** dataset for generating apparels.

14 Hours

Unit 4:Reinforcement Learning, **Diffusion Models, Federated Learning and Overview of Latest Deep Learning Models:**

Introduction, Basic Framework of RL, Learning to Optimize Rewards, Credit Assignment Problem, Temporal Difference, Learning and Q Learning. **Deep RL:** Deep Q Learning, Training and Testing.

Diffusion Model, Stable diffusion architectures, Introduction to Vision Transformers, GPT Architecture. **Federated Learning:** Horizantal, Vertical and FTL (Federated Transfer Learning).

Hands-on : Flower Architecture for MIMC3 dataset **OR** Learning to play a simple game using Double Deep (DD) Q-learning implementation on OpenAI gym.

14 Hours

Tools/ Languages: Pytorch.

Text Book(s):

1: “Advanced Deep Learning with Python” - Ivan Vasilev, Packt Publishing, 2019.
2: “Neural Network and Deep learning” by Charu C Agarwal. Springer International Publishing 2018.

Reference Book(s):

1: “Hands-on Machine Learning with Scikit-Learn and TensorFlow”, Aurelian Geron, O’REILLY, 1st Edition, 2017.
2: “Deep Learning with Keras”, Antonio Gulli and Sujit Pal, Packt Publishing, 1st Edition, 2017.
3: “Pattern Recognition and Machine Learning”, Christopher Bishop, Springer, 1st Edition, 2011 (Reprint).
4:Handouts: Transfer Learning / Latest Deep Learning Techniques / Vision Transformers/ GPT / FL

UE22CS343BB3 - Database Technologies(4-0-0-4-4)

The last decade saw enormous advancements in the design of large-scale data processing systems due to the rise of Internet services. This course covers the architecture of modern data management systems. Topics include storage management, query optimization, distributed, parallel and stream processing, with a focus on the key design ideas shared across many types of data intensive systems.

Course Objectives:

- Data Storage Techniques, Indexing Mechanisms.
- Query Processing and Query Optimization Techniques.
- Parallel and Distributed Databases.
- Data Streaming Systems, current Trends in Design and Implementation of Database Systems.

Course Outcomes:

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

- Design and Deploy Storage Solutions and Indexing mechanisms for optimized performance of databases.
- Perform Query Optimization.
- Apply Parallel and Distributed Database approaches to solve problems of large databases.
- Select an approach to manage "Data Streams" and apply techniques learnt to Develop or Select Specialized Database Applications.

Desirable Knowledge: UE22CS351A – Database Management Systems.

Course Content:

Unit 1- Relational Data Model and Storage Formats and Indexing

Review of Relational Design Theory – Relational Algebra. Secondary Storage Management: The Memory Hierarchy, Flash Storage and Database Buffer, Architecture, Accelerating Access to Secondary Storage, Disks and Disk Failures, Arranging Data on Disk, Block and Record Addresses, Variable-Length Data and Records – Column Stores, Record Modifications. Index Structures: Concepts, Types of Indexes, B+ Tree Indexes, Hash Tables, Multidimensional Data and Indexes and It's Applications, Multiple-key Indexes and R-trees, Bitmap Indexes, Indexes in SQL.

14 Hours

Unit 2 - Query Processing and Optimization

Query Execution: Physical-Query-Plan Operators, One-Pass Algorithms and Two-Pass Algorithms (based on Sorting and Hashing), Clustered and Non-clustered indexes, Buffer Management. The Query Compiler: Parsing and Preprocessing, Algebraic Laws for Improving Query Plans, From Parse Trees to Logical Query Plans, Cost based Plan Selection, Choosing an Order for Joins.

14 Hours

Unit 3- Parallel and Distributed Databases

Avenues for Parallelism: Parallel Database Architectures, Models of Parallelism, Intra-query and Inter-query Parallelism, Parallel Algorithms on Relations. Distributed Databases – Distributed Data Storage, Types, Advantages, Data Fragmentation and Replication, Distributed Transactions, Distributed Query Processing, Distributed Commit – Two-phase Commit, Distributed Locking, Distributed Catalog Management, Concurrency Control, Peer-to-Peer Distributed Search – Chord Circles.

14 Hours

Unit 4 - Data Stream Management & Trends in Design and Implementation of Database Systems

Introducing Stream Processing, Stream Processing Model, Streaming Architectures, Apache Spark as a Stream Processing Engine, Spark's Distributed Processing Model, Spark's Resilience Model, Apache Kafka, Amazon Kinesis. Introduction to Vector database, serverless Vector databases, Use Cases. Current Trends in Design and Implementation of Database Systems and Decision Support Systems, Data Warehousing and Data Mining Applications, Introducing Data Lake, Data Mesh and Data Fabric, Multi-model Databases.

14 Hours

Text Book(s):

1: "Database Systems: The Complete Book", H Garcia-Molina, JD Ullman and J Widom, 2nd Ed., Pearson, 2018.
2: "Stream Processing with Apache Spark", Gerard Maas & Francois Garillot, O'Reilly, June 2019.

Reference Book(s):

1: "Fundamentals of Database Systems", Elmasri and Navathe, Pearson Education, 7th Ed., 2016.
2: "Streaming Systems" by Tyler Akidau, Slava Chernyak, Reuven Lax, O'Reilly, July 2018.

UE22CS343BB4:Machine Learning on Graphs (4-0-0-4-4)

This course introduces the students to the three meta landscapes of representation, optimization, and objective for the learning tasks on graph data. Graphs are ubiquitous in almost everything today –in social graphs, large-scale biological systems, road networks, polypharmacy, and user-item relations. Graphs, being non-Euclidean, require different machine-learning approaches. This course starts by exploring graph features for traditional machine learning on graphs and discussing a few noteworthy classical graph ML approaches. The second unit focuses on shallow embedding methods for graphs in general and then the same for knowledge graphs. The third unit focuses on representation learning using graph neural networks and discusses key algorithms along with extensions like temporal graphs and multi-relational graphs. The fourth unit discusses the deep generative models on the graph and model-building techniques using graph neural networks. The theory input concludes with a discussion on the interdisciplinary applications of Graph Neural Networks, interpretability, and meta-learning perspectives of graph AI.

Course Objectives:

- Learn graph feature-based traditional machine learning.
- Learn shallow embedding approaches for graphs and multi-relational knowledge graphs.
- Learn neural methods of representation learning on graphs.
- Learn GNN modeling techniques and deep graph generative models with interdisciplinary applications.

Course Outcomes:

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

- Should be able to do feature engineering-based graph ML tasks.
- Should be able to do downstream machine learning tasks on graphs using shallow embedding.
- Should be able to solve a downstream machine-learning task using a graph neural network.
- Should be able to use graph modeling techniques and appreciate deep generative models on graphs.
-

Desired Knowledge: UE22CS352A – Machine Learning, UE22CS343AB3 -Graph Theory and its applications

Course Content:

Unit 1: Traditional machine learning on graph

Types of complex Graphs and computational tasks on Graphs. Graph Features for traditional machine learning– Node level features. Graph kernel features- Graphlet & Motif, Weisfeiler-Lehman kernel. Measures for neighborhood overlap. Graph clustering and spectral methods –Graph Laplacians, Graph Cuts and clustering, Spectral clustering. Encoder-Decoder perspective – encoder, decoder, and optimization of the encoder-decoder model. Factorizationbased approaches.

14 Hours

Unit 2: Graph Representation Learning Based on shallow embedding methods

Random Walk-based embedding –DeepWalk and Node2Vec. Limitations of shallow embedding. Shallow embedding in multi-relational Knowledge Graph- embedding as reconstruction task, loss function, and decoders. Knowledge Graph Completion Task-TransE, TransR, DistMult, ComplEx. Semi-supervised Learning on Graph - Label Propagation Algorithm and Label Spreading Algorithm

14 Hours

Unit 3: Graph Learning using graph neural network

Vanilla GNN - Neural message passing framework. Generalized neighborhood aggregation. Generalized update methods. Graph Convolution Networks (GCN): Spectral methods(GCN, AGCN) and Spatial Methods(DCNN, DGCN, LGCN and GraphSAGE), Multi-relational GCN and Graph Attention Networks (GAT), Graph Transformer. Graph pooling. Expressiveness, designing Maximally expressive GNNs, Expressive Power by aggregate functions. Applications and loss functions.

14 Hours

Unit 4: Deep Generative Models, GNN on complex graphs and interdisciplinary applications

Deep Generative models on graph – Variational autoencoder on the graph, Generative adversarial network on the graph, Auto-regressive methods. GNN on complex graphs – Heterogeneous GNN, Dynamic GNN, Hypergraph GNN. Application of GNN to Biology and Chemistry, Interdisciplinary Applications of GNN: Unstructured data – Text and Image. Structured Data - Social Network, Recommender System, Cyber Security. An introduction to trustworthy graph AI, Meta-Learning perspectives of graph ML.

14 Hours

Tools/Languages: NetworkX for statistical features of graphs, Tensorflow, Keras, and Scikit Learn for traditional graph ML, and Pytorch Geometric for Graph Neural Networks.

Text Book(s):

1: “Graph Representation Learning”, William L Hamilton, Morgan and Claypool Publishers, 2020.
2: “Deep Learning on graphs”, Yao Ma and Jiliang Tang, Cambridge University Press, 2021
3. “Introduction to Graph Neural networks”, Zhiyuan Liu and Jie Zhou, Synthesis Lectures on Artificial Intelligence and Machine learning, Morgan and Claypool Publishers, 2020.

Reference Book(s):

1: “Social Media Mining”, Reza Zafarani, Cambridge University Press, 2015 (Asian Economic edition available)
2: “ Network Science” by Albert Barabasi , 2016

UE22CS343BB6 : Information Security (4-0-0-4-4)

This course will present security aspects from a secure software life cycle process – requirement, architecture, design, coding, and testing. Students will have opportunity to dwell well in to technical "how to" with hands-on sessions, assignments, and some case study discussions.

Course Objectives:

- To understand various cyber threats and attacks and secure software development process.
- To learn attack and defence mechanisms for buffer overflow, shellshock attack, etc.
- To understand the concept of threat modelling and its application.
- To learn about the most common web application security vulnerabilities. and apply various penetration testing techniques and tools.

Course Outcomes:

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

- Identify possible misuse cases in the context of software development.
- Defend against various attacks and how to write secure code.
- Apply threat modelling techniques to expose inherent vulnerabilities in applications.
- Design and develop secure web applications and Exploit software vulnerabilities and launch attacks.

Course Content:

Unit 1: Introduction and Privilege Escalation Attacks

Software Threats, Attacks and Vulnerabilities, CIA Triad, OWASP Top 10, CVE, Security and reliability, Security vs. privacy, Cyberattack Types, Anatomy of an Attack, Security Concepts and Relationships. Use cases and Misuse cases, Misuse case legend, Security use case vs Misuse case, Secure Software Development Life Cycle (SDL). Case Study: Target case study. Set-UID program: Need for privileged programs, Set-UID mechanism, Superman story, Attack surfaces, Invoking other surfaces, Principle of least privilege. Environment variables and attacks: Environment variables, Attack surface, Attacks via Dynamic linker, External program, and Library. Lab: Set-UID program & Environment variables and attacks. Shellcode attack on Set-UID and CGI programs.

14 Hours

Unit 2: Software Vulnerabilities and Malicious Software

Buffer overflow attack: Program memory layout, Stack and function invocation, Stack buffer-overflow attack, Attacks with Unknown address and Buffer size, Shellcode, Countermeasures & Defeating it. Return-to-libc attack: Introduction, Launch the attack part I & part II. Format string vulnerability: Introduction to functions and format string, Vulnerable program, Exploiting the vulnerability, Code injection attack, Countermeasures. **Case study: Target case study**. Malware and its Types, Malware analysis: Conifer, Morris, Stuxnet worm, Ransomware.

14 Hours

Unit 3: Threat modelling and Basic Web Security

Threat Modelling, Trust Boundaries, Attack Surfaces, Brainstorming, Modelling Methods, STRIDE model and variants, Defensive tactics, and Technologies, Privacy Threats, Taxonomy and Types. Web security basics, Attacks on HTTP GET and POST services, Cross Site Request Forgery (XSRF/CSRF): Cross-site requests and its problems, CSRF attacks, Countermeasures. **Case study: Apple - Privacy vs Safety.**

14 Hours

Unit 4: Web application security and Penetration Testing

(XSS/CSS) Attack: CSS attack, CSS attacks in action, Self-propagation, Preventing CSS attacks. SQL injection attack: Introduction to SQL, interacting with database in web, Launching SQL injection attacks, Countermeasures. Static analysis, Penetration testing: Introduction, Benefits, Drawbacks, Penetration testing tools and Fuzzing, Patching.

14 Hours

Note: Hands-on experience for relevant topics in the form of Lab and/or Assignment is given. Relevant cyber security case study for undergraduate students is discussed.

Hands-on exercises:

1. Set-UID program & Environment variables and attacks
2. Shellcode attack
3. Buffer overflow attack
4. Return-to-libc attack.
5. Format string vulnerability.
6. CSS/XSS attack
7. CSRF attack.
8. SQL injection.

9. Shell shock attack.
10. Penetration testing.(Assignment)
11. Malware 1,2,3 (Assignment)
12. Case study 1,2 (Target, Apple)

Tools / Languages: SEED Labs VM, Scapy, Burp Suite, Metasploit, Nmap, etc.

Text Book(s):

1: “Computer & Internet Security: A Hands-on Approach”, Wenliang Du, 2nd Edition/3rd Edition.

Reference Book(s):

1: “Computer Security: Principles and Practice”, William Stallings and Lawrie Brown, Pearson Education, 3rd Edition, 2014.
2: “Secure Programming with Static Analysis”, Brian Chess and Jacob West, Pearson Education, 2007

UE22CS343BB7: Mobile and Autonomous Robots (4-0-0-4-4)

This course introduces students to the field of autonomous mobile robots. Students will learn the basic concepts and techniques necessary for designing, building, and programming autonomous robots, as well as explore the latest developments in this exciting field. The course will include lectures, hands-on activities, and projects.

Course Objectives:

- Understanding the past, present, and future of autonomous mobile robotics. Getting exposed to the current state-of-the-art of scientific literature.
- The course aims to teach the theoretical and practical fundamentals involved in designing and operating autonomous robots or intelligent agents.
- The introductory discussions cover subtopics like robot perception, planning, and control.
- Other major topics include designing robot parts, integrating sensors, analyzing motion kinematics, simulation testing using ROS/ROS2, handling unmodeled environmental and social factors, and preparing for field deployment.

Course Outcomes:

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

- Students will gain a comprehensive understanding of autonomous robots and their applications. Understand the ROS architecture and communication protocols and use ROS packages for robot software control.
- Understand the principles of robot manipulators, end-effectors, and identify the sensors and actuators used in robot control.
- They will be able to use path planning and obstacle avoidance algorithms for robot navigation.
- Ultimately, students will be able to apply this knowledge to design and develop their own autonomous robot systems.

Desirible Knowledge : Robotics with ROS (New Elective Course for Vth sem)

Course Content:

Unit 1: Introduction to Autonomous Robots

Overview of autonomous robots, History and current state of autonomous robots, Basic concepts and terminology, Applications of autonomous robots, Robot Hardware and Software: Sensors and actuators Overview of robot sensors, ROS overview: ROS/ROS2 for robotics, ROS architecture and communication protocols, ROS packages for robot hardware control.

14 Hours

Unit 2: Locomotion and Perception

Ground robots (UGVs) robots, Forward and inverse kinematic, The basics of wheel types and arrangements. Aerial robots (UAVs), Dynamics of thrusters and propellers, The basics of SOTA locomotion systems. Visual and inertial measurements: Gyroscope, accelerometers, IMU, GPS, Range sensors, camera vision and LiDAR.

14 Hours

Unit 3: Robot Vision and Localization

Visual perception: Image processing and filtering, Object detection and tracking, Stereo vision and 3D perception, The basics of scene segmentation and parsing. Introduction and Challenges of Localization, Belief Representation, Map representation, Probabilistic Map-Based Localization: Markov localization, Kalman Filter localization, SLAM: Simultaneous localization and mapping, EKF SLAM, Particle filter SLAM, GraphSLAM, MonoSLAM, Open challenges in SLAM.

14 Hours

Unit 4: Navigation

Introduction Path Planning and Navigation, Path planning: Graph search and Potential field path planning. Obstacle avoidance: Bug algorithm, Vector field histogram, bubble band technique, Curvature velocity techniques, Dynamic window approaches. Revisiting Navigation using Reinforcement Learning and Imitation learning: A Robotics Perspective, Applications and Social Implications.

14 Hours

Text Book(s):

1: Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents series) second edition by Roland Siegwart, Illah Reza Nourbakhsh, Davide Scaramuzza

2: Probabilistic Robotics By Sebastian Thrun, Wolfram Burgard and Dieter Fox. ISBN-13: 978-0262201629, ISBN-10: 0262201623. Intelligent Robotics and Autonomous Agents series; 1st Edition

Reference Book(s):

1: Introduction to Autonomous Robots: Nikolaus Correll, Magellan Scientific, 2016.
2: ROS Robot Programming, ROBOTIS Co., Ltd. From the basic concept to practical programming and robot application. YoonSeok Pyo, HanCheol Cho, RyuWoon Jung, and TaeHoon Lim.
3: Introduction to Robotics: Mechanics and Control 4th Edition, John Craig, ISBN-13: 978-0133489798, Pearson; 4th edition

Recommended Materials
1: Robot Operating System (ROS): http://wiki.ros.org/ROS/Tutorials
2: ROS2 tutorials: https://docs.ros.org/en/foxy/Tutorials.html

UE22CS343BB8 - Security for Internet of Things

This course enables the learner to understand the security threats associated with Internet of Things and perform security testing on connected devices within our homes and enterprise to build a better model for protecting ourselves.

Course Objective:

- Understand the threats of IoT devices
- Understand vulnerabilities, possible attacks and learn the security testing methodology for networks, IoT hardware and radio protocols.
- Enhance knowledge with hands-on experience on IoT security tools and attack analysis.
- Understand the various Security measures to be adopted to secure IoT devices

Course Outcome:

- Identify and describe the variety of IoT systems architectures, essential components and challenges specific to IoT systems.
- Analyze various network and hardware security mechanisms for IoT devices
- Gain hands-on experience on different tools to target IoT ecosystem.
- Analyze and apply appropriate security and privacy solutions for real-world applications.

Course Contents:

Unit 1: Introduction to IoT Security

Introduction to IoT Security, Traditional security vs IoT security,Basic concepts of IoT Architecture from security perspective, Challenges of IoT Security, OWASP Top 10 security risks and consumer IoT security guidance, Threat Modelling for IoT attack, Common IoT threats, Network Hacking: VLAN hopping in IoT networks.

14 Hours

Unit 2: Network Security and Hardware Security for IoT devices

MQTT authentication, Analysing Network protocols: Wireshark dissector and Nmap Scripting Engine module for the DICOM protocol, exploiting zero-configuration Networking: UPnP, mDNS, DNS-SD, and WS-Discovery, Hardware Hacking: UART, JTAG, and SWD Exploitation- Hacking an STM32F103 microcontroller using UART and SWD.SPI and I2C, Firmware Hacking.

14 Hours

Unit 3: Radio Hackingand Smart IoT devices hacking

Radio Hacking:Short Range Radio: Abusing RFID, Bluetooth low energy, Medium Range Radio: Hacking Wi-Fi, Long Range Radio: LPWAN, Smart home, Hacking the smart home: Gaining physical entry to a building, cloning a keylock system's RFID Tag, Jamming the Wireless alarm.

14 Hours

Unit 4: Targeting the IoT Ecosystem and Secure Design of IoT Devices

Playing back an IP Camera stream, Analysing IP Camera Network traffic, Extracting the video stream, attacking a Smart treadmill, Secure design goals: Mitigate automated attack risks, Secure points of integration, Hardware protection measures, IoT IAM infrastructure: PKI for IoT, Revocation support: OCSP, SSL pinning, Authorization and access control with OAuth 2.0 Cryptographic Controls for IoT Protocols.

14 Hours

Tools and Languages : Wireshark, Yersinia, VoIP Hopper, Bettercap,aircrack-ng

Text Book(s):

1: "Practical IoT Hacking", Fotios Chantzis, Ioannis Stais, Paulino Calderon, Evangelos Deirmentzoglou, Beau Woods, March 2021, No Starch Press Publishers, ISBN: 9781718500907.

Reference Book(s):

1: "Practical Internet of Things Security" , Brian Russell, Drew Van Duren, Packt Publishers, 2nd Edition

UE22CS343BB8 :Applied ML in IoT with TinyML

Course Outcomes:

- Design and deploy simple Machine Learning models on microcontrollers for various IoT applications.
- Preprocess sensor data, optimize code, and evaluate the trade-off between model accuracy and resource consumption.
- Utilize tools like TensorFlow Lite Micro and TinyML Kit for TinyML development.
- Integrate sensor data acquisition, communication protocols, and data pipelines for real-time TinyML applications.

Course Objectives:

By the end of this course, students will be able to:

- Understand the fundamental concepts of ML, Deep Learning (DL), and their application in the IoT domain.
- Utilize TensorFlow Lite Micro and TinyML Kit for developing and deploying ML models on microcontrollers.
- Design and implement TinyML applications for various functionalities like keyword spotting, image classification, and sensor fusion.
- Evaluate and interpret the performance of TinyML models on real hardware.

Unit-1: Foundations of Machine Learning: Introduction to IoT & TinyML, The ML paradigm, Building blocks of DL, NN, Regression with Dense NN, Classification with Dense NN, Image Classification using CNN, Datasets and Model Performance Metrics, Preventing Overfitting.

Unit-2: TensorFlow Lite for Microcontrollers & TinyMLKit: What Is TensorFlow Lite for Microcontrollers?, Build Systems, Supporting a New Hardware Platform, Supporting a New IDE or Build System, Integrating Code Changes Between Projects and Repositories, Contributing Back to Open Source, Supporting New Hardware Accelerators, Understanding the File Format, Porting TensorFlow Lite Mobile Ops to Micro, TinyML Kit Overview, TinyML Kit Setup, TinyML Kit Sensor Testing, Sensor Testing, Sensor Fusion.

Unit-3: Applications and Deployment to Microcontrollers (MCUs)-1: TF-Lite, TFL-Micro, TFL-Micro Hello-World example, KeyWord Spotting (KWD) using Edge Impulse, KDW dataset creation, Micro Speech Example, Workflow, Model development & testing, Person-detection example, Image classification

Unit-4: Applications and Deployment to Microcontrollers (MCUs)-2: Magic Wand: Building and Application, Training a Model, Building a Weather Station with TensorFlow Lite for Microcontrollers, Voice Controlling LEDs with Edge Impulse.

Tools: Arduino Nano 33 BLE sense , Arduino IDE , Google Colab

Text Books:

[1] Pete Warden and Daniel Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, O'Reilly Media, 2020.

[2] Vijay Janapa Reddi at Harvard and open-source collaborators, Machine Learning Systems with TinyML, Open-source collaborative-effort book, 2023-present.

[3] Gian Marco Iodice and Ronan Naughton, TinyML Cookbook: Combine artificial intelligence and ultra-low-power embedded devices to make the world smarter, 2022.

[4] Francois Chollet, Deep Learning with Python, Manning, Second Edition, 2021.