UE20CS101 : Python for Computational Problem Solving (4-0-0-4-4)

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.
- 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.
- 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 .

12 Hours

Unit 2 : Collections & Basics of Functions

Lists, Tuples , Dictionaries, Sets, Strings and text file manipulation: reading and writing files. Functions : Definition, call.

12 Hours

Unit 3 : Functions and GUI

Positional and keyword parameter, Default parameters, Variable number of arguments, Recursion, Callbacks, Closure, Decorators. Graphical User Interface with Tinkter package- Different geometric methods – Tk, mainloop, Creating simple GUI - buttons, canvas, check button, labels, entry fields, dialogs Widgets - sizes, fonts, colors layouts, nested frames.

12 Hours

Unit 4 : Functional Programming, Modules,Testing and Debugging

Map, filter, reduce, max, min. lambda function - list comprehension, Modules - import mechanisms,Testing- Pytest , Function testing with Doctest, pdb debugger commands.

10 Hours

Unit 5 : Object Oriented Programming

Classes and objects - inheritance , polymorphism. Error handling & Exceptions - try, except and raise , exception propagation.

10 Hours

Tools / Languages : Python.

Text Book:

1. “Introduction to Computer Science Using Python: A Computational Problem-Solving Focus” Charles Dierbach,Wiley India Edition,John Wiley, 2015.

Reference Books:
1: “Learn python Programming”,Fabrizio Romano, 2 nd Edition,Packt Publishing,2018.
2: “Fundamentals of Python: First Programs”, Kenneth A.Lambert, Cengage,2019.
3: “Introduction to Computation and Programming Using Python: With Application to Understanding Data”,John V. Guttag,MIT Press,MIT with Library of Congress Cataloging- in-Publication Data,2016.

UE20CS102 : Python for Computational Problem Solving Laboratory (0-0-2-1-1)

This laboratory mainly focus on solving the problem(s) using python data structures such as list, tuple, strings, sets and dictornary),functions and files. This allows the students to develop computational problem solving skills.

Course Objectives:

- Learn basics of computer programming.
- Learn how to solve a given problem.
- Learn to use various paradigms of programming.
- Learn to test and debug python code.

Course Outcomes:

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

- Illustrate problem solving using Python programming.

Course Content:

1. UNIX Commands and Utilities.
2. Program to demonstrate Input Output Functions, Operators and Expressions.
3. Program to demonstrate the Usage of Libraries and Control Structures(Selection statements).
4. Program to demonstrate Control Structures (Loops).
5. Program to demonstrate Lists and Tuples.
6. Program to demonstrate Sets and Dictionaries.
7. Program to demonstrate String Related Operations.
8. Program to demonstrate File Handling in Python.
9. Program to demonstrate the Usage of Functions.
10. Program to demonstrate Functional Programming.
11. Program to demonstrate Functional Programming and pdb debugger.
12. Program to demonstrate classes and objects.
13. Programs to demonstrate exception handling.
14. Project.

Tools / Languages : Python.

Reference Book(s):

Laboratory Manual prepared by the Department of Computer Science and Engineering, PES University.

UE20CS151 : Problem Solving with C (4-0-0-4-4)

In Problem Solving with C Course, will be learning to solve common types of computational problems, by analyzing the given problem statement and developing an algorithm to solve the given problem.Then make use of c language constructs to develop well-structured programs.

Course Objectives:

- Learn how to solve common types of computing problems.
- Learn to map problems to programming features of ‘C’.
- Understand computer programming and its roles in problem solving.
- Understand and develop well-structured programs using ‘C’ language.
- Learn the basic data structures through implementation in ‘C’ language.

Course Outcomes:

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

- Analyze the given problem and develop an algorithm to solve the problem.
- Optimize the solution given for an existing problem.
- Use ‘C’ language constructs in the right way.
- Design, develop and test programs written in ‘C’.

Course Content:

Unit 1 : Counting

Introduction to Programming, Salient Features of ‘C’, Program Structure, Variables, Data Types, Operators and Expressions, Control Structures, Input/ Output Functions.

10 Hours

Unit 2 : Text Processing and String Manipulation Single Character Input and Output, Arrays and Pointers, Strings, String Manipulation.

12 Hours

Unit 3 : Prioritized Scheduling

Functions, Structures and Unions, Dynamic Memory Management, Lists, Priority Queue.

12 Hours

Unit 4 : Sorting

Sorting, Combination of Structures, Arrays and Pointers, Callback, Sorting using Callback.

12 Hours

Unit 5 : File Handling

File Handling, Enums, Bit Fields, Storage Class, Qualifiers, Life and Scope, Pre-Processor Directives, Conditional Compilation, Pragmas.

10 Hours

Tools / Languages : C

Text Book:

1.“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 Puzzle Book” by Alan R. Fever, Pearson Education.
3: “Expert C Programming: Deep C Secrets” by Peter Van Der Linden, Pearson Education.

UE20CS152 : Problem Solving With C Laboratory (0-0-2-1-1)

In Problem Solving with C Course, will be learning to solve problems by writing pseudocode first and then followed by programming by making use of C constructs and then test for possible required test cases.

Course Objectives:

- Learn and implement how to solve common types of computing problems.
- Use data types and control structures of ‘C’.
- Learn to map problems to programming features of ‘C’.
- Learn to write good, portable ‘C’ programs.

Course Outcomes:

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

- Analyze a given problem and implement an algorithm to solve the problem.
- Improve upon a solution to a problem.
- Implement the ‘C’ language constructs in the right way.
- Design, develop and test programs written in ‘C’.

Course Content:

1. Program to demonstrate Input, Output Functions and Control Structures.
2. Program to demonstrate Word/ Line/ Character Count in a Given Input Data.
3. Program to demonstrate Operators and Control Structures.
4. Program to demonstrate Character Input and Output.
5. Program to demonstrate Functions, Arrays and Pointers.
6. Program to demonstrate Strings, Pointers using Multiple Files Usage.
7. Program to demonstrate the use of Multi-Dimensional Arrays.
8. Program to demonstrate the usage of Structures, Array of Structures and Array of Pointers.
9. Program to demonstrate List using multiple files.
10. Program to demonstrate Enumerations.
11. Program to demonstrate File Handling in ‘C’.
12. Program to demonstrate File Handling in ‘C’.

Tools / Languages : C

Reference Book(s):

Laboratory Manual prepared by the Department of Computer Science and Engineering, PES University.

UE19CS201: Digital Design and Computer Organization (4-0-0-4-4)

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.
- Use of studied digital building blocks to construct more complex systems.

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 datapath 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.
- Utilize the concepts and techniques learnt to implement complex digital systems.

Course Content:

Unit 1 : Combinational Logic Design

Introduction, Boolean Functions, Truth tables, Boolean Algebra, Identities, Logic minimization, K-Maps, Adder/Subtractor, Overflow.

12 Hours

Unit 2 : Combinational and Sequential Logic Design

Muxes, Decoders, Shifters, Gate/Wire delays, Timing, Latches, Flip-flops, Synchronous logic design, Finite State Machines.

12 Hours

Unit 3 : Sequential and Arithmetic Circuits

FSM examples, Counters, Memory arrays, Carry-lookahead and Prefix adders.

10 Hours

Unit 4 : Arithmetic Circuit and Architecture

Shift/add multiplier/divider, Wallace tree multiplier, Floating point, Assembly Language, Machine Language.

10 Hours

Unit 5 : Microarchitecture

Addressing Modes, Performance analysis, Single-cycle, Multi-cycle processor datapath and control, Systolic array matrix multiply, Overview of computer systems organization.

12 Hours

Text Book:

1. “Digital Design and Computer Architecture”, David Money Harris, Sarah L Harris, 2nd Edition, Morgan Kaufmann, 2012.

Reference Book(s):

1: “Digital Design”, M Morris Mano, Michael D Ciletti, 6th Edition, Pearson, 2018.
2: “Computer Organization and Design”, David A Patterson, John L Hennessey, 5th Edition, Elsevier, 2016.

3: “Computer Organization and Design”, Carl Hamacher, Safwat Zaky, Zvonko Vranesic, 5th Edition, Tata McGraw-Hill,2011.

UE19CS202 : Data Structures and its Applications (4-0-0-4-4)

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.
- Construct essential skills of data structures to store and retrieve data quickly and usefully (efficiently).
- Usage the 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 skills required to implement any data structure.
- Minimal data structure that supports all operations needed.
- Appropriate data structure in competitive programming.
- Design and develop efficient software systems with good knowledge of data structures.

Course Content:

Unit 1 : Overview

Static and Dynamic Memory Allocation, Singly Linked List.

Linked List: Doubly Linked List, Circular Linked List – Single and Double, Multilist : Introduction to sparse matrix (structure). **Application:** Case Study **-Text Editor , Assembler-** Creation of a Symbol Table.

12 Hours

Unit 2 : Stacks and Queues

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. **Queues & Deque:** Basic Structure of a Simple Queue, Circular Queue, Priority Queue, Deque and its implementation using Arrays and Linked List. **Applications of Queue:** Case Study – Josephus problem, CPU scheduling- Implementation using queue(simple /circular).

12 Hours

Unit 3 : Trees and Heaps

General : N-ary trees, Binary Trees, Binary Search Trees and Forest: definition, properties, conversion of an N-ary tree and a Forest to a binary tree. Implementation of BST using arrays and dynamic allocation : Insertion and deletion operations, Traversal of trees: Preorder, Inorder and Postorder. 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** (Words with their meanings), Simulation of an Airport(Flight landing & takeoff using priority queues implemented by heaps).

12 Hours

Unit 4 : Balanced Trees and Graphs

Balanced Trees: definition, AVL Trees, Rotation.

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. Application of BFS and DFS: Connectivity of graph, finding path in a network. Case Study – Indexing in databases(B Tree: K-way tree)- Insertion and deletion operations with examples.

10 Hours

Unit 5 : Suffix Tree and Hashing

Suffix Trees: Definition, Introduction of Trie Trees, Suffix trees.Implementation of TRIE trees, insert, delete and search operations. **Hashing:** Simple mapping / Hashing: hash function, hash table, Collision Handling: Separate Chaining & Open Addressing, Double Hashing, Rehashing. **Applications:** Cryptography, URLs decoding,Word prediction using TRIE trees / Suffix Trees

10 Hours

Tool/ Language : ‘C- Language’

Text Book:

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

Reference Book:

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

UE19CS203 : Statistics for Data Science (4-0-0-4-4)

Data Science is the study of data. It is about extracting, analyzing, visualizing, managing and storing data to create insights. This course covers both Descriptive statistics to understand the data and the Inferential statistics which seeks to infer something about a population on the basis of a statistical sample and build a simple linear regression model.

Course Objectives:

- Provide insights about the basic roles of a Data Scientist. Develop a greater understanding of the importance of Data Visualization techniques.
- Provide students with knowledge of Random Variables and Distributions of it.
- Provide students with knowledge of Confidence Intervals and its importance.Make inferences about the population parameters using sample data.
- Make inferences about the population parameters using sample data and test it to draw meaningful conclusions.
- Provide an understanding on the importance and techniques of predicting a relationship between the two sets of data and determine the goodness of fit model.

Course Outcomes:

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

- Use Python and other tools to extract, clean and analyze data from several data sources (files, web) and analyze an extremely large data set and perform exploratory data analysis to extract meaningful insights.
- Analyze a real-world problem and solve the same with the knowledge gained from various distributions study.
- Compute Confidence Intervals.
- Develop and test a hypothesis about the population parameters to draw meaningful conclusions.
- Fit a regression model to data and use it for prediction.

Course Content

Unit 1 : Introduction to Data Science

Introduction, **Sampling:** Sampling Methods, Sampling Errors. **Getting and Analyzing Data:** Reading Files, Scraping the Web, Need for Data Cleaning and its Basics. **Statistics:** Introduction, Types of Statistics. **Data Visualization and Interpretation:** Histogram, Bar Charts, Box Plots, Scatter Plots, Heat Maps, Good vs. Bad Visualization,Case Study.

12 Hours

Unit 2 : Random Variables and Probability Distributions

Random Variables, Expectation, Functions of Random Variables. **Probability Distributions:** Discrete Distributions (Bernoulli, Binomial, Poisson), Continuous Distributions (Normal), Derivation of Distributions, Generation of Random Variates, Case Study.

12 Hours

Unit 3 : Confidence Intervals

Principles of Point Estimation - Mean Squared Error, Maximum Likelihood Estimate, The Central Limit Theorem and Applications, Normal Probability Plots.Confidence Intervals: Using Simulation to Construct Confidence Intervals, Interval Estimates for Mean of Large and Small Samples, Interval Estimates for Proportion of Large Samples, Confidence Intervals for the Difference between Two Means, Interval Estimates for Paired Data. Factors affecting Margin of Error,Case Study.

10 Hours

Unit 4 : Hypothesis and Inference

Hypothesis Testing for Population Mean and Population Proportion of Large Samples, Drawing conclusions from the results of Hypothesis tests, Large sample tests for a Population proportion, Large - Sample tests for

Difference between two means, Distribution Free Tests, Chi-squared Test, Fixed Level Testing, Type I and Type II Errors, Case Study.

12 Hours

Unit 5 : Power of Test and Simple Linear Regression
Power of a Test, Factors Affecting Power of a Test. Simple Linear Regression: Introduction, Correlation, the Least Square Lines, Predictions using regression models - Uncertainties in Regression Coefficients, Checking Assumptions and transforming data,Case Study. .

10 Hours

Tools / Languages : Python**.**

Text Book:

1. "Statistics for Engineers and Scientists", William Navidi, McGraw Hill Education, India, 4th Edition, 2015.

Reference Book:
1: "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling", Raj Jain,Wiley, 2008.
2: "Sampling- Design and Analysis" ,Sharon L. Lohr, 2nd edition (stats), Cengage, 2010.
3: "Data Science from Scratch", Joel Grus, O'Reilly, 1st Edition, 2015.
4: "Statistics for Engineers and Scientists", William Navidi, 3rd Edition, McGraw Hill, 2010.

UE19CS204 : 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 learn building blocks of a website using HTML, CSS and JavaScript.
- The core concepts of HTML5, Advanced JavaScript, JQuery and AJAX through simple hands-on exercise.
- The industry trending MERN (**M**ongoDB, **E**xpressJS, **R**eactJS and **N**odeJS) stack and build an UI of the application using React JS.
- The binding of an UI to the MongoDB through NodeJS.
- Middleware Express JS and learn 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 richer user experience by implementing HTML5 features and Asynchronous communication through JQuery.
- Understand industry standard applications using MERN stack layers (**M**ongoDB, **E**xpressJS, **R**eactJS, **N**odeJS) and apply the features of React JS for a rich User Interface.
- Integrate the UI with MongoDB database through NodeJS.
- Create RESTful Web services using ExpressJS.

Course Content:
Unit 1 : HTML, CSS and Client Side Scripting
Introduction to WWW and Web protocols, HTTP Request Response Formats, URLs, Basic Mark-ups & syntax, HTML elements & attributes, Web Form 2.0 and Form Controls, CSS3.0-Styles and Style sheets, Selectors, Style properties, Box Model, JavaScript Basics(variables, scope and lexical operator (fat Arrow)), Javascript objects.

10 Hours

Unit 2 : HTML5, JQuery And Ajax
DOM Manipulations, Events and Event Handling in JavaScript, XML Vs JSON, HTML5 (New Tags, New Inputs, Elements & API), JQuery Introduction, Callbacks and Promises, Single Page Application, Asynchronous Communication.

12 Hours

Unit 3 : ReactJS
MERN Introduction, React installation and application setup with web pack, JSX, React Classes and Components, Rendering of elements, Properties, State, Context, Component lifecycle methods, Refs & Keys, Event Handling, React Router, Stateless components, React form & controls.

12 Hours

Unit 4 : MongoDB and NodeJS
Understanding Node JS Architecture, NPM Installation and Features, Set up Node JS app, HTTP Methods and Verbs, query string, call backs , buffers, streams, File system, MongoDB-Documents, Collections, Reading and Writing to MongoDB, MongoDB NodeJS Driver, Running a react application on NodeJS.

10 Hours

Unit 5 : ExpressJS
Introduction to Web services and REST API's , Express Installation and Server setup, Building the application stack, Routing, List API, Create API, Error Handling, Express Middleware, Express Scaffolding and Templates.

12 Hours

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

Text Book:
1. "Learning PHP, MySQL & JavaScript", Robin Nixon. May 2018,5th edition, O'Reilly Media, Inc. ISBN: 9781491978917
2. "Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node", by Vasan Subramanian. March 2017, Apress

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.

UE19CS205 : 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. The course covers the notions of Finite State Automaton, Regular expression, Push Down Automaton, Context Free Languages 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.
- To familiarize students to construct Regular Expressions, 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 familiarize students with concepts like Recursively Enumerable languages, Recursive Languages, Undecidable Problems.
- 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, 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 are not, 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 Undecidable Problem to another, Undecidable Problems for Recursively Enumerable Languages.
- 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.

10 Hours

Unit 2 : Regular Languages and Grammars

Regular Expressions Connection between Regular Expressions and Regular Languages Regular Grammars. **Properties of Regular Languages:** Closure Properties of Regular Languages, Elementary Questions about Regular Languages, Identifying Non Regular Languages.

10 Hours

Unit 3 : Pushdown Automata and Context Free Languages

Definitions of PDA and CFL, Deterministic Pushdown Automata, Non-Deterministic Pushdown Automata, Pushdown Automata and Context Free Languages, Context Free Grammars, 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.

12 Hours

Unit 4 : Properties of Context-Free Languages

Properties of Context-Free Languages: Closure Properties and Questions about Context–Free Languages, Pumping Lemma for Context–Free Languages. **Turing Machines:** The Standard Turing Machine, Constructing Turing Machines, Combining Turing Machines for Complicated Tasks, Turing’s Thesis. **Hierarchy of Formal Languages and Automata:** Recursive and Recursively Enumerable Languages, The Chomsky Hierarchy. **Limits of Algorithmic Computation:** Some Problems that cannot be solved by Turing Machines, Undecidable Problem for Recursively Enumerable Languages.

12 Hours

Unit 5 : Propositional Logic

A very simple Logic, Syntax, Semantics, A simple knowledge Base, A simple inference procedure. **Propositional Theorem 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.

12 Hours

Tools / Languages : JFLAP.

Text Book:

1. “An Introduction to Formal Languages and Automata”, Peter Linz, Jones and Bartlett, New Delhi, India, 5th Edition, 2011.
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.

UE19CS206: Digital Design and Computer Organization Laboratory (0-0-2-1-1)

DDCO Lab essentially consists of implementation and simulation of a series of digital building blocks in an HDL (Hardware Description Language). The developed blocks are finally composed to yield a simple microprocessor.

Course Objectives:

- Explain the elements of digital system abstractions such as digital representations of information, Digital Logic, Boolean Algebra, State Elements and Finite State Machine (FSMs).
- Design simple digital systems based on these digital abstractions, using the "Digital Paradigm" including discrete sampled information.
- Use the "Tools of the Trade" - Basic Instruments, Devices and Design Tools.
- Work in a design team that can propose, design, successfully implement and report on a digital systems project.
- Communicate the purpose and results of a design project in written and oral presentations.

Course Outcomes:

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

- Achieve knowledge and awareness of various components to design stable digital circuits.
- Analyze and design combinational circuits.
- Design and develop sequential circuits.
- Design and develop a basic microprocessor.
- Translate real world problems into digital logic formulations using Verilog.

Course Content:

1. Hardware Kit Assignment.
2. Verilog Basics – I.
3. Verilog Basics – II.
4. Mux, Adder Design.
5. ALU .
6. Register File .
7. Datapath.
8. Program Counter.
9. Control Logic.
10. Microprocessor.
11. Project Assignment (microprocessor based).

Tools / Languages : Icarus Verilog Simulator, GTKWave Waveform Viewer.

Reference Book:

1. Laboratory Manual prepared by the Department of Computer Science and Engineering, PES University.

UE19CS207: Data Structures and its Applications Laboratory (0-0-2-1-1)

This laboratory focuses on concrete implementations of basic data structures such as Stacks, Queues, Linked lists, trees and graphs to be implemented in variety of ways. This allows the student to appreciate the choices and tradeoffs which face a programmer in a real situation.

Course Objectives:

- Basic approaches and mindsets for analyzing and designing data structures.
- Construct essential skills of data structures to store and retrieve data quicklyand usefully (efficiently).
- Usage the 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, Graph.
- Implement a given application using the available data structure.

Course Outcomes:

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

- Relevant data structures for any given application.
- Apply skills required to implement any data structure.
- Minimal data structure that supports all operations needed.
- Appropriate data structure in competitive programming.

Course Contents:

1. Implement Singly Linked list and perform all types of insert & delete, display operations.
2. Implement Doubly Linked list and perform all types of insert & delete, display operations .
3. Implement Singly Linked Circular list and Doubly linked Circular list and perform insert, delete, display operations .
4. Implement operations of on Stack & its applications.
5. Implement operations of a Queue, Circular queue.
6. Implement Binary search tree and perform the operations of insert, delete and traverse the BST.
7. Implement priority Queue using heap(both min and max).
8. Implementation of TRIE tree.
9. Implementation of HASH function, Hash table. Use rehashing and double hashing techniques to resolve collision
10. Week #10 to Week #14: Implementation of case studies.

Tools / Languages : Hacker Earth , C – Language.

Reference Book:

1. Laboratory Manual prepared by Department of Computer Science and Engineering, PES University.

UE19MA251 : Linear Algebra (4-0-0-4-4)

This is an undergraduate course in engineering. Linear algebra is the study of linear systems of equations, vector spaces, and linear transformations. This course will concentrate on the mathematical theory and methods of linear algebra.

Course Objective :

- Understand matrix operations to solve systems of linear equations and find the inverse of a matrix.
- Understand the basic concepts of vector spaces, linear transformations and fundamental subspaces.
- Understand the concept of orthogonality of vectors & subspaces and Gram Schmidt orthogonalization to produce orthonormal vectors.
- Understand the concept of eigenvalues, eigenvectors for diagonalization, Singular value decomposition which is used to compute Pseudo inverse, Least squares fitting of data, Multi variable control and Matrix approximation.
- Use Scilab to solve problems related to visualize the various concepts of Linear Algebra.

Course Outcomes:
At the end of the 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.
- Analyze linear transformation as a mapping and calculate its matrix representation with respect to standard and nonstandard bases.
- Understand the concepts of orthogonal vectors and orthogonal subspaces and apply the Gram-Schmidt process to find an orthonormal basis in a subspace.
- Analyze eigenvalues , eigenvectors , largest eigen value and diagonalization of a matrix.
- Understand the concept of Positive definite matrices, Singular ValueDecomposition and its applications.

Course Content:
Unit 1 : Matrices and Gaussian Elimination
Introduction, The Geometry of Linear Equations, Gaussian Elimination, Singular cases, Elimination Matrices, Triangular factors and Row Exchanges, Inverses and Transposes, Inverse by Gauss -Jordan method.
Self Learning Component: Algebra of Matrices.
Case Study: Parallelism of Gaussian Elimination, Simple Cryptography application using Inverse Matrix.
11 Hours
Unit 2 : Vector Spaces
Vector Spaces and Subspaces (definitions only) , Linear Independence, Basis and Dimensions, The Four Fundamental Subspaces.
Self Learning Component: Examples of vector spaces and subspaces.
Case Study: Data transmission (Hamming code error detection and error correction), Solving Linear Homogenous Differential Equations.
12 Hours
Unit 3 : Linear Transformations and Orthogonality
Linear Transformations , Orthogonal Vectors and Subspaces, Cosines and Projections onto Lines, Projections and Least Squares.
Self Learning Component: Inner Products and Cosines.
Case Study: Illustration of Image Editing using Linear Transformation, Linear Regression.
11 Hours

Unit 4 : Orthogonalization , Eigen Values and Eigen Vectors
Orthogonal Bases, The Gram- Schmidt Orthogonalization, Introduction to Eigen values and Eigen vectors, Properties of Eigen values and Eigen vectors, Power Method to find the Largest Eigen Value, Diagonalization of a Matrix.
Case Study: Feature Selection using Eigen Vectors , Power of a Matrix.

12 Hours

Unit 5: Singular Value Decomposition
Symmetric Matrices and Quadratic Forms, Tests for positive definiteness, Positive Definite Matrices and Least Squares, Semi definite Matrices, Singular Value Decomposition, Applications of the SVD.
Case study: SVD Image compression technique.

10 Hours

Tools/ Languages : SciLab, Python.

Text Book:

1. "Linear Algebra and its Applications", Gilbert Strang, 4th Edition, Thomson Brooks/ Cole, Second Indian Reprint 2007.

Reference Book(s):
1: "Linear Algebra and its Applications" , David .C.Lay, 3rd Edition,2002.
2: "Higher Engineering Mathematics" , B S Grewal,, Khanna Publishers, 44th Edition, 2017.

SCILAB:

- Introduction and Gaussian Elimination
- Inverse of a Matrix by the Gauss- Jordan Method
- The LU Decomposition
- The Span of Column Space of a Matrix
- The Four Fundamental Subspaces
- The Gram-Schmidt Orthogonalization
- Projections by Least Square
- Eigen values and Eigen Vectors of a Matrix
- The Largest Eigen Value of a Matrix by the Power Method.

Reference Book(s):
1: "Numerical Methods: Principles, Analysis, And Algorithms", S. Pal, Oxford University Press , 1st Edition, 2009.
2: "Numerical Methods", E. Balaguruswamy , Tata McGraw - Hill Education, New Delhi 1st Edition, 1999.

UE19CS251 : 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 endeavor 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 to design and analyze algorithms with an emphasis on the resource utilization in terms of time and space.
- Learn various techniques in the development of algorithms so that the effect of problem size and architecture design on the efficiency of the algorithm is appreciated.
- Learn to apply appropriate algorithmic design techniques for specific problems.
- Learn to trade space for time in algorithmic design using input enhancement and per-structuring.
- Learn to improve the limitations of algorithmic power.

Course Outcomes:

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

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

Course Content:

Unit 1 : Introduction

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. Algebraic structures - Rings, Fields and Groups.

12 Hours

Unit 2 : Brute Force and Divide-and-Conquer

Brute Force: Selection Sort, Bubble Sort, Sequential Search, Brute Force String Matching, Exhaustive Search. **Divide-and-Conquer:** Master Theorem, Merge Sort, Quick Sort, Binary Search, Binary Tree Traversals, Multiplication of Large Integers, Strassen's Matrix Multiplication.

12 Hours

Unit 3 : Decrease-and-Conquer & Transform-and-Conquer

Decrease-and-Conquer: Insertion Sort, Topological Sorting, Algorithms for Generating Combinatorial Objects, Decrease-by-a-Constant-Factor Algorithms. **Transform-and-Conquer:** Pre-sorting, Heap Sort, Red-Black Trees, 2-3 Trees and B Trees.

10 Hours

Unit 4 : Space and Time Tradeoffs & Greedy Technique

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.

10 Hours

Unit 5 : Limitations & Coping with the Limitations of Algorithm Power

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. **Dynamic Programming:** Computing a Binomial Coefficient, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms.

12 Hours

Tools / Languages : GCC Compiler.

Text Book:
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, 2/e, 2007.
3: “Algorithm Design”, Jon Kleinberg, Eva Tardos, Pearson Education, 2006.

UE19CS252 : Microprocessor and Computer Architecture (4-0-0-4-4)

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.

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.
- Introduction to I/O Systems, interface and interaction with processor.
- 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.

Pre-requisite Courses: UE19CS201- 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.

12 Hours

Unit 2 : Pipeline

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

12 Hours

Unit 3 : Memory Hierarchy

Basics of Caches - Fully Associative, Direct Mapped and Set Associative, Cache Performance, Basic Cache Optimizations.

12 Hours

Unit 4 : I/O and Bus Architecture

Accessing I/O Devices, I/O Device Interface, PCI, SCSI, USB, Program Controlled I/O, DMA, Synchronous and Asynchronous Bus, AMBA and ASB Bus.

10 Hours

Unit 5 : Advances in Architecture

Introduction to Parallel Architecture, , Types of Parallelism, Amdahl’s Law, Gustafson Law, HardwareMulti threading, Multi-Core Architecture.

10 Hours

Text Book(s):

1. “Computer Organization and Design”, Patterson, Hennessey, 5th Edition, Morgan Kaufmann,2014.
2. “ARM System-on-Chip Architecture”, Steve Furber, 2nd Edition, 2015, Pearson India.

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.

UE19C253 : Computer Networks (4-0-0-4-4)

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 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 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 and study the IPv4 and IPv6 protocols and addressing.
- 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.
- Explain basic concepts of application layer protocols like DNS, HTTP and implement simple client-server applications using socket programming.
- Understand the concept of reliable and unreliable data transfer protocols and how TCP and UDP implement these concepts.
- Demonstrate the ability to configure the routers and services such as DHCP, ICMP and NAT and implement logical addressing schemes.
- 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

Introduction to Computer Networks - What is the Internet? - A Nuts-and-Bolts Description - A Services Description - What is a Protocol? - The Network Edge: Access Networks - Physical Media – 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 – Introduction to Cloud Computing.

10 Hours

Unit 2 : Application Layer

Network Application Principles: Network Application Architectures, Processes Communication, Transport Services available to Applications, Transport Services provided by Internet; 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.

12 Hours

Unit 3 : Transport Layer

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; 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.

12 Hours

Unit 4 : Network Layer and Internet Protocol

Overview of Network Layer: Forwarding and routing, Network Service Models; Inside Router: Input port processing and destination-based forwarding, Switching, Output port processing, where does Queueing occur? Packet scheduling; The Internet Protocol (IP) IPV4: Datagram Format, Fragmentation, Addressing, NAT; Introduction to Network layer Protocols: DHCP, ICMP; IPv6 Addressing: Address space allocation, Global unicast addresses, Auto configuration, Renumbering, IPv6 Protocol: Packet Format, Transition from IPv4 to IPv6; Introduction to Routing Algorithms: Link State and Distance Vector.

12 Hours

Unit 5 : Link Layer and LAN

Link layer – Error-Detection and Correction techniques, Parity checks, Internet checksum, cyclic redundancy check, Multiple access protocols: CSMA/CD; 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, Analog vs Digital Signals, Transmission media. Wireless LANs: IEEE 802.11 LAN architecture, 802.11 MAC Protocol, IEEE 802.11 Frame.

10 Hours

Tools/ Languages: Wireshark, Python.

Text Book:

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.

2: "Mastering Cloud Computing: Foundations and Applications Programming", Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, Morgan Kaufmann, 2013.

UE19CS254 : 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.

Course Objectives:

- The course focuses on fundamental operaing system concepts
- The course provides an understanding of various components of operating system.
- The course delve deeper into various algorithms and associated tradeoffs for efficient resource management such as process, disk and memory management.
- The course will introduce design principles and tradeoffs in the design of Operating Systems.
- The course will also introduce the concepts such as security, protection and virtualization.

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

- Gain extensive knowledge on principles and modules of Operating Systems.
- Understand the design of various algorithms for scheduling and their relative performance.
- Design pieces of the operating systems such as process management, concurrent processes and threads, memory management, virtual memory.
- Use the tools and the interface of the operating system.
- Explore design tradeoffs in designing various components of the Operating system.

Pre-requisite Courses:UE19CS202-Data Structures and its Applications.

Course Content:
Unit 1 : Introduction and Process Management
What Operating Systems Do, Computer-System Organization, Computer-System Architecture, Operating-System Structure & Operations, Kernel Data Structures,Computing Environments, Operating-System Services, OperatingSystem Design and Implementation. Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination, CPU Scheduling and Scheduling Algorithms, IPC – Shared Memory & Message Passing, Pipes – Named and Ordinary. Case Study: Linux/Windows Scheduling Policies.

12 Hours

Unit 2 : Threads and Concurrency
Introduction to Threads, types of threads, Multicore Programming, Multithreading Models, Thread creation, Thread Scheduling, PThreads and Windows Threads, Mutual Exclusion and Synchronization: software approaches, principles of concurrency, hardware support, Mutex Locks, Semaphores. Classic problems of Synchronization: Bounded-Buffer Problem, Readers -Writers problem, Dining Philosophers Problem concepts. Synchronization Examples - Synchronisation mechanisms provided by Linux/Windows/Pthreads. Deadlocks: principles of deadlock, tools for detection and Prevention.

12 Hours

Unit 3 : Memory Management
Main Memory: Hardware and control structures, OS support, Address translation, Swapping, Memory Allocation (Partitioning, relocation), Fragmentation, Segmentation, Paging, TLBs context switches Virtual Memory – Demand Paging, Copy-on-Write, Page replacement policy – LRU (in comparison with FIFO & Optimal), Thrashing, design alternatives – inverted page tables, bigger pages. Case Study: Linux/Windows Memory.

12 Hours

Unit 4 : Storage Management
Mass-Storage Structure – Mass-Storage overview, Disk Scheduling, Swap-Space Management, RAID structure. File System Interface - file organization/structure and access methods, directories, sharing. File System Implementation/Internals: File control Block (inode), partitions & mounting, Allocation methods.

Case Study: Linux/Windows File Systems.

10 Hours

Unit 5 : I/O Management and Security

I/O Hardware, polling and interrupts, DMA, Kernel I/O Subsystem and Transforming I/O Requests to Hardware Operations - Device interaction, device driver, buffering. System Protection: Goals, Principles and Domain of Protection, Access Matrix, Access control, Access rights. System Security: The Security Problem,Program Threats, System Threats and Network Threats. Case Study: Windows 7/Windows 10.

10 Hours

Tools/ Languages : Pthread, Experimental Academic OS.

Text Book:

1. "Operating System Concepts" , Abraham Silberschatz, Peter Baer Galvin, Greg Gagne 9th Edition, John Wiley & Sons, 2013.

Reference Book(s):

1: "Operating Systems, Internals and Design Principles" , William Stallings, 9th Edition, Pearson, 2018 .
2: "Operating Systems": Three Easy Pieces, RemziArpaci-Dusseau and Andrea ArpaciDusseau, http://pages.cs.wisc.edu/~remzi/OSTEP/ .
3: "Advanced Programming in the Unix Environment", Richard Stevens and Stephen A Rago, Pearson, 3rd edition,2017.
4: "Operating Systems", Harvey Deitel, Paul Deitel, David Choffnes, 3rd Edition, Prentice Hall, 2004.
5: "Modern Operating Systems" , Andrew S Tanenbaum, 3rd edition, Pearson,2007.

UE19CS255 : Computer Networks Laboratory (0-0-2-1-1)

Computer Networks Laboratory helps students learn how to put "principles into practice," in a hands-on-networking lab course. These labs will be done in a networked lab setting with ethernet switches, NICs, desktops and cables. Cisco Packet Tracer, a visual simulation tool designed by Cisco Systems allows students to create network topologies and imitate modern computer networks. Claynet, a network virtualization tool helps students realize large scale networks using routing and switching NFV technology.

Course Objectives:

- Learn the basic network utilities to analyse and troubleshoot networks.
- Sniff, filter, and analyze network traffic with Wireshark.
- Understand the header fields and their values of HTTP, TCP, UDP, IP and IEEE 802.11 protocols.
- Study how protocols and layering are represented in packets.
- Apply the client-server model in network socket programming.

Course Outcomes:

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

- Be familiar with the network simulator Packet Tracer.
- Assess different solutions for computer networks.
- Understand the functionality of seven layers of OSI reference model.
- Realize network communication skills through socket programming.
- Design and build a small sized wired or wireless LAN.

Course Contents:

1. Basic network command-line utilities to monitor/analyse/troubleshoot networks.
2. Understand the building blocks of networks and usage of Cisco Packet Tracer & Claynet network virtualization platform with reference to OSI Layer.
3. Network traffic/protocol analysis using packet sniffer – Wireshark.
4. Network performance effects of HTTP persistent and non-persistent connections.
5. HTTP protocol investigation on HTTP message formats, Cookies, Conditional Get, Authentication and Caching.
6. Setting up a DNS server to understand the functionality and its operations.
7. Extensive network socket programming for applications: Multi-threaded web server, E-mail client, UDP Pinger, Web proxy server.
8. Analyzing and troubleshooting transport and network layer protocols like TCP, UDP and DHCP using Wireshark.
9. Understand IPv4 addressing and static routing (Desktops & Claynet).
10. Understand IPv6 addressing and static routing (Claynet).
11. Understanding ICMP redirect messages, Broadcast storm and TTL expiry in L2 network (Desktops & Claynet)
12. Analyzing and troubleshooting Layer 2 protocols (MAC, Ethernet frames, ARP)
13. Building and testing a small network using Cisco Packet Tracer.
14. Investigation on 802.11 wireless network protocol.

Tools / Languages : Claynet, Wireshark, Cisco Packet Tracer.

Reference Book(s):

1: Laboratory Manual prepared by Department of CSE, PES University.

UE19CS256 : Microprocessor and Computer Architecture Laboratory(0-0-2-1-1)

This Lab is to complement the theory learnt. This lab will help in understanding the low level programming needs of a microprocessor and understanding the performance issues of system architecture in the context of a complex hardware system including pipeline and memory hierarchy. This is understood using suitable ools for Low level programming design, analysis, architecture simulation, implementation, and debugging.

Course Objectives:

- Implement assembly language programs and develop strong competencies in contemporary ISAs.
- Develop, edit, compile and debug assembly language programs using present - day simulators.
- Know various addressing modes that are defined in a given instruction set architecture and illustrate how machine language instructions in that architecture identify the operand(s) of each instruction.
- Practice interfacing experiments using various sensors with an Arduino board.
- Learners to imbibe the skills of formulation of a complex problem, design a suitable solution using Arduino/ Raspberry Pi processors and demonstrate the end results.

Course Outcomes:

- Inculcate the importance of instruction set architecture and their fundamental concepts using assembly language programming.
- Demonstrate editing, compiling, executing and debugging an assembly language program of a contemporary microprocessor.
- Demonstrate the usage of subroutines and recursion supported by the ISA.
- Imbibe strong assembly language programming skills by implementing solutions to problems using simulators.
- Instilling the idea to formulate a complex problem definition, approach to solve the problem, methodology to apply and implement suitable algorithm and check for the final results.

Course Content:

1. Introduction to Instruction Set – ARM Processor. Sample programs using Simulator
2. Programs on ARM/ x86 using Simulator.
3. Programs on ARM/ x86 using Simulator.
4. Demonstration of Arduino Microcontroller with various sensors, Project Discussion.
5. Project Discussion and project Title confirmation.
6. Programs on ARM/ x86 using Simulator.
7. Introduction to 3 stage Pipeline using simulator and Pluggins.
8. Introduction to 5 stage Pipeline (case study: Hazards using Simulator - RAW, WAR, WAW).
9. Cache Memory Demonstration.
10. Mini Project
11. Mini Project
12. Mini Project
13. Project Evaluation

Tools / Languages : ARM Simulator, Ardino Microcontroller kit, MIPS pipeline simulator, ParaCache simulator.

Reference Book(s):
1: The ARMSim User Guide.
2: 5-stage MIPS simulator online.
3: Para cache simulator online.
4. Laboratory Manual prepared by Department of CSE, PES University.

UE18CS301 : Computer Networks (4-0-0-4-4)

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 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 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 and study the IPv4 and IPv6 protocols and addressing.
- 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.
- Explain basic concepts of application layer protocols like DNS, HTTP and implement simple client-server applications using socket programming.
- Understand the concept of reliable and unreliable data transfer protocols and how TCP and UDP implement these concepts.
- Demonstrate the ability to configure the routers and services such as DHCP, ICMP and NAT and implement logical addressing schemes.
- 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

Introduction to Computer Networks - What is the Internet? - A Nuts-and-Bolts Description - A Services Description - What is a Protocol? - The Network Edge: Access Networks - Physical Media – 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 – Introduction to Cloud Computing.

10 Hours

Unit 2 : Application Layer

Network Application Principles: Network Application Architectures, Processes Communication, Transport Services available to Applications, Transport Services provided by Internet; 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.

12 Hours

Unit 3 : Transport Layer

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; 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.

12 Hours

Unit 4 : Network Layer and Internet Protocol
Overview of Network Layer: Forwarding and routing, Network Service Models; Inside Router: Input port processing and destination-based forwarding, Switching, Output port processing, where does Queueing occur? Packet scheduling; The Internet Protocol (IP) IPV4: Datagram Format, Fragmentation, Addressing, NAT; Introduction to Network layer Protocols: DHCP, ICMP; IPv6 Addressing: Address space allocation, Global unicast addresses, Auto configuration, Renumbering, IPv6 Protocol: Packet Format, Transition from IPv4 to IPv6; Introduction to Routing Algorithms: Link State and Distance Vector.

12 Hours

Unit 5 : Link Layer and LAN
Link layer – Error-Detection and Correction techniques, Parity checks, Internet checksum, cyclic redundancy check, Multiple access protocols: CSMA/CD; 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, Analog vs Digital Signals, Transmission media. Wireless LANs: IEEE 802.11 LAN architecture, 802.11 MAC Protocol, IEEE 802.11 Frame.

10 Hours

Tools/ Languages: Wireshark, Python.

Text Book:
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.

2: “Mastering Cloud Computing: Foundations and Applications Programming”, Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, Morgan Kaufmann, 2013.

UE18CS302 : 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.

Course Objectives:

- The course focuses on fundamental operaing system concepts
- The course provides an understanding of various components of operating system.
- The course delve deeper into various algorithms and associated tradeoffs for efficient resource management such as process, disk and memory management.
- The course will introduce design principles and tradeoffs in the design of Operating Systems.
- The course will also introduce the concepts such as security, protection and virtualization.

Course Outcomes:

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

- Gain extensive knowledge on principles and modules of Operating Systems.
- Understand the design of various algorithms for scheduling and their relative performance.
- Design pieces of the operating systems such as process management, concurrent processes and threads, memory management, virtual memory.
- Use the tools and the interface of the operating system.
- Explore design tradeoffs in designing various components of the Operating system.

Pre-requisite Courses:UE18CS202-Data Structures, UE18CS253 - Microprocessor and Computer Architecture.

Course Content:

Unit 1 : Introduction and Process Management

What Operating Systems Do, Computer-System Organization, Computer-System Architecture, Operating-System Structure and Operations, Kernel Data Structures,Computing Environments, Operating-System Services, OperatingSystem Design and Implementation. Process concept: Process in memory, Process State, Process Control Block, Process Creation & Termination, CPU Scheduling & Scheduling Algorithms, IPC – Shared Memory and Message Passing, Pipes – Named & Ordinary. Case Study: Linux/Windows Scheduling Policies.

12 Hours

Unit 2 : Threads and Concurrency

Introduction to Threads, types of threads, Multicore Programming, Multithreading Models, Thread creation, Thread Scheduling, PThreads and Windows Threads, Mutual Exclusion and Synchronization: software approaches, principles of concurrency, hardware support, Mutex Locks, Semaphores. Classic problems of Synchronization: Bounded-Buffer Problem, Readers -Writers problem, Dining Philosophers Problem concepts. Synchronization Examples - Synchronisation mechanisms provided by Linux/Windows/Pthreads. Deadlocks: principles of deadlock, tools for detection and Prevention.

12 Hours

Unit 3 : Memory Management

Main Memory : Hardware and control structures, OS support, Address translation, Swapping, Memory Allocation (Partitioning, relocation), Fragmentation, Segmentation, Paging, TLBs context switches Virtual Memory – Demand Paging, Copy-on-Write, Page replacement policy – LRU (in comparison with FIFO and Optimal), Thrashing, design alternatives – inverted page tables, bigger pages.
Case Study: Linux/Windows Memory.

12 Hours

Unit 4 : Storage Management

Mass-Storage Structure – Mass-Storage overview, Disk Scheduling, Swap-Space Management, RAID structure. File System Interface - file organization/structure and access methods, directories, sharing File System Implementation/Internals: File control Block (inode), partitions and mounting, Allocation methods.

Case Study: Linux/Windows File Systems.

10 Hours

Unit 5 : IO Management and Security

I/O Hardware, polling and interrupts, DMA, Kernel I/O Subsystem and Transforming I/O Requests to Hardware Operations - Device interaction, device driver, buffering System Protection: Goals, Principles and Domain of Protection, Access Matrix, Access control, Access rights. System Security: The Security Problem,Program Threats, System Threats and Network Threats. Case Study: Windows 7/Windows 10.

10 Hours

Tools/ Languages : Pthread, Experimental Academic OS.

Text Book:

1. “Operating System Concepts” , Abraham Silberschatz, Peter Baer Galvin, Greg Gagne 9th Edition, John Wiley & Sons, 2013.

Reference Book(s):

1: “Operating Systems, Internals and Design Principles” , William Stallings, 9th Edition, Pearson, 2018 .
2: “Operating Systems”: Three Easy Pieces, RemziArpaci-Dusseau and Andrea ArpaciDusseau, http://pages.cs.wisc.edu/~remzi/OSTEP/ .
3: “Advanced Programming in the Unix Environment”, Richard Stevens and Stephen A Rago, Pearson, 3rd edition,2017
4: “Operating Systems”, Harvey Deitel, Paul Deitel, David Choffnes, 3rd Edition, Prentice Hall, 2004.
5: “Modern Operating Systems” , Andrew S Tannenbaum, 3rd edition, Pearson, 2007.

UE18CS303: Machine Intelligence (4-0-0-4-4)

Machine Intelligence (MI) 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 familiarise students with the breadth of modern MI, to impart an understanding of the dramatic surge of MI in the last decade, and to foster an appreciation for the distinctive role that MI can play in shaping the future of our society.

Course Objectives:

- Familiarize the concepts of Intelligent Agents and Search Methods.
- 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, Nature based Optimization and Computational Theory.

.**Course Outcomes:**

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

- Apply Intelligent Search methods for a variety of problems.
- Distinguish categories of Data Attributes, Dimensions, Sample Sizes.
- Acquire a thorough understanding of Supervised, Unsupervised Learning, Ensemble Methods and Nature based Optimization.
- Extract Associations and Rules, provide impactful recommendations from data.
- Design and Analyse machine learning algorithms using computational learning theory.

Pre-Requisite: UE18CS203-Introduction to Data Science, UE18MA251-Linear Algebra, UE18CS252-Design and Analysis of Algorithms

Course Content:

Unit 1 : Introduction, Classification with Decision trees and Performance Metrics

Introduction to AI and ML , Intelligent Agents and its Types, Machine Learning and its Models, Problem solving by Searching- Uninformed Search and Informed Search Methods, Perspectives and Issues, designing learning systems, Concepts of hypotheses, Version space, inductive bias, Performance metrics-accuracy, precision, recall, sensitivity, specificity, AUC, ROC, Bias Variance decomposition. Decision Trees- Basic algorithm (ID3), Hypothesis search and Inductive bias, Issues in Decision Tree Learning – Overfitting, Solutions to overfitting, dealing with continuous values

12 Hours

Unit 2 : Supervised Learning with KNN, ANN, SVM

Instance-based learning: k-nearest neighbour learning, Artificial Neural networks: Introduction, Perceptrons, Multi-layer networks and back-propagation, Activation Units, Support Vector Machines – margin and maximization, SVM - The primal problem, the Lagrangian dual,SVM – Solution to the Lagrangian dual.

12 Hours

Unit 3 : Boosting and Stochastic Models

Improving performance with Ada-boost, combining weak learners. Bayesian Learning – Bayes theorem, Concept learning, Maximum likelihood,Bayes optimal classifier, Naïve Bayes classifier, Expectation maximization and Gaussian Mixture Models, Hidden Markov Models.

10 Hours

Unit 4 : Unsupervised Learning and Dimensionality Reduction

Hierarchical vs. non-hierarchical clustering, Agglomerative and divisive clustering, K-means clustering, Bisecting k-means, K-Means as special case of Expectation Maximization, Apriori algorithm - Association analysis, the Apriori principle. Finding frequent itemsets, mining association rules, FP-growth – FP trees, Mining frequent items from an FP-Tree, Dimensionality reduction techniques – PCA, SVD.

12 Hours

Unit 5 : Genetic Algorithms and Computational Learning Theory

Overview: Genetic Algorithms – Representing hypothesis, Genetic operators and Fitness function and selection,. Introduction to PSO and application in Single Objective optimization problems, Computational Learning Theory, PAC-Learnability, The Vapnik-Chervonenkis Dimension

10 Hours

Tools / Languages : Tensorflow 1.15, Keras 2.3.1, Python 3.7.

Text Books:

- “Artificial Intelligence: A Modern Approach (3rd Edition)”, Stuart Russel and Peter Norvig, Pearson , 2009.
- “Machine Learning”, Tom Mitchell, McGraw Hill Education (India), 2013.

Reference Book(s):

1: “Machine Learning The Art and Science of Algorithms that Make Sense of Data”, Peter Flach,,Cambridge University Press(2012).

2: “Pattern Recognition and Machine Learning”, Christopher Bishop, Springer (2nd Printing), 2011.

UE18CS304 : Computer Networks Laboratory (0-0-2-1-1)

Computer Networks Laboratory helps students learn how to put "principles into practice," in a hands-on-networking lab course. These labs will be done in a networked lab setting with ethernet switches, NICs, desktops and cables. Cisco Packet Tracer, a visual simulation tool designed by Cisco Systems allows students to create network topologies and imitate modern computer networks. Claynet, a network virtualization tool helps students realize large scale networks using routing and switching NFV technology.

Course Objectives:

- Learn the basic network utilities to analyse and troubleshoot networks.
- Sniff, filter, and analyze network traffic with Wireshark.
- Understand the header fields and their values of HTTP, TCP, UDP, IP and IEEE 802.11 protocols.
- Study how protocols and layering are represented in packets.
- Apply the client-server model in network socket programming.

Course Outcomes:

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

- Be familiar with the network simulator Packet Tracer.
- Assess different solutions for computer networks.
- Understand the functionality of seven layers of OSI reference model.
- Realize network communication skills through socket programming.
- Design and build a small sized wired or wireless LAN.

Course Content:

- Basic network command-line utilities to monitor/analyse/troubleshoot networks.
- Understand the building blocks of networks and usage of Cisco Packet Tracer & Claynet network virtualization platform with reference to OSI Layer.
- Network traffic/protocol analysis using packet sniffer – Wireshark.
- Network performance effects of HTTP persistent and non-persistent connections.
- HTTP protocol investigation on HTTP message formats, Cookies, Conditional Get, Authentication and Caching.
- Setting up a DNS server to understand the functionality and its operations.
- Extensive network socket programming for applications: Multi-threaded web server, E-mail client, UDP Pinger, Web proxy server.
- Analyzing and troubleshooting transport and network layer protocols like TCP, UDP and DHCP using Wireshark.
- Understand IPv4 addressing and static routing (Desktops & Claynet).
- Understand IPv6 addressing and static routing (Claynet).
- Understanding ICMP redirect messages, Broadcast storm and TTL expiry in L2 network (Desktops & Claynet).
- Analyzing and troubleshooting Layer 2 protocols (MAC, Ethernet frames, ARP).
- Building and testing a small network using Cisco Packet Tracer.
- Investigation on 802.11 wireless network protocol.

Tools / Languages : Claynet, Wireshark, Cisco Packet Tracer.

Reference Book(s):

1: Laboratory Manual prepared by Department of CSE, PES University.

UE18CS305: Operating Systems Laboratory (0-0-2-1-1)

This Lab includes hands-on exercises to familiarize important tools, utilities, system calls, program using shell scripting in a UNIX based environment as well as implementation of system calls and modules for process, memory and resource management towards better understanding of OS design and implementation.

Course Objectives:

- Familiarize with important tools, utilities and shell programming in Linux/Unix based Environment.
- Introduce various interfaces of the operating system, including system calls.
- Understand the structure of a process, how it is created & scheduled and synchronization between processes/threads.
- Understand concepts of memory management such as paging and persistent storage.
- Understand the design of an operating system and process to modify parts of the operating system.

Course Outcomes:

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

- Create processes, threads and synchronize between processes/threads.
- Modify, compile and boot a standalone operating system.
- Design and implement process scheduling algorithms.
- Implement new system calls.
- Design files system and memory management modules in an Operating system at an introductory level.

Course Content:

1. Learn to use UNIX/Linux commands related to Process creation, Process monitoring, Linux File system tree, Process states and Linux File system commands including pipes and filters.
2. Shell scripting – 1.
3. Shell scripting – 2.
4. System Calls: Build simple Client-Server program to transfer file from server to client using system calls open (), read (), write (), close (), fork(), exec(), wait() .
5. Process: Build a simple Shell to transfer a file from the server and pipe it through a word count program.
6. Threads: Build a Multi-Threaded Server to transfer a file from server to client using Producer Consumer Model using Pthreads.
7. Program to solve synchronization problem between threads using pthreads .
8. Xv6 Operating System: Exploring, compiling and booting Xv6.
9. Xv6 System Calls: To add a new system call to Xv6. Add command line to see it.
10. Xv6 Process Scheduling: Modify Process Scheduler to take into account the new process priority and schedule accordingly.
11. Xv6 Process Management: Implementing corner cases in the Process Management to ensure no process starvation to occur.
12. Xv6 Memory Management: Exploring Memory Management in Xv6.
13. Xv6 File System: Understanding File System structure on Xv6.

Reference Book(s):

1. Laboratory Manual prepared by the Department of Computer Science and Engineering, PES University.

UE18CS311: 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.

Course Objectives:

- Enable the learner with basics of Amortized Complexity Analysis of Data Structures.
- Empower the learner with String Matching/ Prediction Algorithms.
- 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.
- 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.

Pre-Requisite:UE18CS251 – 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.

12 Hours

Unit 2 : String Algorithms

Naïve 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.

10 Hours

Unit 3 : Maximum Flow, Polynomials and FFT

Flow Networks, The Ford-Fulkerson method, The Edmonds-Karp algorithm, Maximum Bi-Partite Matching, Polynomials and FFT: Representation of Polynomials, Efficient Polynomial Multiplication, DFT and FFT, Efficient Implementation of FFT.

12 Hours

Unit 4 : 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.

10 Hours

Unit 5 : 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.

12 Hours

Tools / Languages : C/C++.

Text Book:

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.

UE18CS312 : 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.

Course Objectives:

- Understand and apply data pre-processing, summarization and visualization concepts pertaining to the analysis of data.
- Build, evaluate and validate analytical models using various techniques.
- Analyze data to infer underlying patterns and formulate recommendations.
- Understand the role of data analytics in business decision making.

Course Outcomes:

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

- Perform exploratory data analysis on a given set of data using effective data visualization techniques.
- Build regression and time series models for prediction.
- Be able to build classification models and cluster text data.
- Develop intelligent decision support systems using analytical models built.

Pre-Requisite: UE18CS203 - Introduction to Data Science.

Course Content:

Unit 1 : Exploratory Data Analysis and Visualization

Introduction, Data Sources, Data Cleaning, Dimensionality Reduction, Data Summarization, Visualization – Graphics and Plotting (R Graphics and relevant packages and maps), Case Studies.

10 Hours

Unit 2 : Regression Analysis

Linear regression, Non-linear regression, Multivariate Regression, Logistic Regression, Ridge Regression, Lasso Regression, Case Studies.

12 Hours

Unit 3 : Time Series Analysis

Decomposing a time series, Simple and exponential smoothing, ACF/ PACF, concept of stationarity – its importance, testing for stationarity and converting a non-stationary signal to stationary, AR, MA, ARMA and ARIMA modelling, Case Studies.

12 Hours

Unit 4 : Recommendation Systems

Collaborative filtering, Knowledge-based filtering: decision trees and agglomerative clustering with a mention of KNN, SVM and DBSCAN (including a brief introduction to text classification/ clustering), Mining of association rules and evaluation of recommendation systems, Case Studies.

12 Hours

Unit 5 : Advanced Techniques

Sparse PCA, Latent Semantic Analysis, Discrete Markov Chain, Confounding variables, Case Studies.

10 Hours

Tools / Languages : R and Python.

Text Book:

1. "Business Analytics, The Science of Data-Driven Decision Making", U. Dinesh Kumar, Wiley 2017.

Reference Book(s):

1: “Data Mining: Concepts and Techniques” by Jiawei Han, Micheline Kamber and Jian Pei, The Morgan Kaufmann Series in Data Management Systems, 3rd Edition, 2011.
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.

UE18CS313 : Internet of Things (4-0-0-4-4)

The explosive growth of the “Internet of Things” is changing our world – both at Home and at work. In this course, students will learn about the various building blocks of IoT, the interfaces with the Edge/Cloud, IoT related protocols for Data communication and delve into IoT design considerations, constraints and trade-off’s. Students will round off by executing a project

Course Objectives:

- Learn the fundamentals of Internet of Things.
- Learn about smart objects and IoT network architecture.
- Compare different Application protocols for Internet of Things.
- Familiarize the Importance of Data Analytics and Security in IoT.
- Apply IoT for real word entities and role of IoT in various domains of Industry.

Course Outcomes:

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

- Apply the fundamentals of Internet of Things.
- Build a small low-cost embedded system using Arduino / Raspberry Pi or equivalent boards.
- Evaluate different infrastructure components and network systems, and design the basic network for IoT solutions.
- Employ the need of Data Analytics and Security in IoT.
- Analyse applications of Internet of Things in real world scenario Provide lab session for each unit to help gain deeper inside to IoT.

Pre-Requisite: UE18CS151-Problem solving with C.

Course Content:

Unit 1 : Introduction

What is IOT? Trends in adoption of IOT, Convergence of IT and IOT, Challenges in IOT, IOT network Architecture and design, physical design and logical design, Drivers behind New Network Architectures, Comparing IoT Architectures, A Simplified IoT Architecture, The Core IoT, IOT design Methodology, Domain specific IOT, Functional Stack, IoT Data Management and Compute Stack.

12 Hours

Unit 2 : Smart objects

The “Things” in IoT, Sensors, Actuators, and Smart Objects, Sensor Networks, Connecting Smart Objects, Communications Criteria, IoT Access, Technologies, IoT platforms, Programming with Arduino, Programming with Raspberry Pi and Node MCU.

12 Hours

Unit 3 : IP as the IoT Network Layer

The Business Case for IP, The need for Optimization, Optimizing IP for IoT, Profiles and Compliances, Application Protocols for IoT, The Transport Layer, IoT Application Transport Methods, Networking technologies, Communication aspects Wireless medium access issues, Common protocols, MQTT, Software & Management Tools for IoT.

12 Hours

Unit 4 : Data and Analytics for IoT

Data Analytics for IoT,Data acquisition and preparation, Machine Learning, Big Data Analytics Tools and Technology, Edge Streaming Analytics, Network Analytics, Design considerations to secure hardware Data analytics, Connecting IoT to cloud – Cloud Storage for Iot Securing IoT, A Brief History of OT Security, Common Challenges in OT Security, How IT and OT Security Practices and Systems Vary, Formal Risk Analysis Structures: OCTAVE and FAIR, The Phased Application of Security in an Operational Environment, Identify and analyze IoT security, Privacy risks.

10 Hours

Unit 5 : Case Studies and Advanced Topics

IoT Physical Devices and Endpoints - Arduino UNO: Introduction to Arduino, Arduino UNO, Fundamentals of Arduino Programming. IoT Physical Devices and Endpoints – Raspberry Pi: Introduction to Raspberry Pi, About the Raspberry Pi Board: Hardware Layout, Operating Systems on Raspberry Pi, Programming Raspberry Pi with Python, Wireless Temperature Monitoring System Using Pi, Connecting Raspberry Pi via SSH, Accessing Temperature from DS18B20 sensors, Remote access to Raspberry Pi, Introduction to ESP32 Dev Board , Programming ESP32 with Arduino, Smart and Connected Cities, An IoT Strategy for Smarter Cities, Smart City IoT Architecture, Smart City Security Architecture, Smart City, Home automation, Industry applications, Surveillance applications, Rural IoT, Various Real time applications of IoT.

10 Hours

Tools / Languages : Arduino IDE.

Text Book:

1. "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, 1st Edition, Pearson Education (Cisco Press Indian Reprint). (ISBN: 978-9386873743), 2017.

Reference Book(s):

1:"Internet of Things – A hands-on approach",Arshdeep Bahga, Vijay Madisetti, Universities Press, 2015.
2: "Designing the Internet of Things", Adrian McEwen, Hakin Cassimally Publisher Wiley 2013.
3: Enterprise IoT by Dirk Slama, Frank Puhlmann, Jim Morrish, Rishi M Bhatnagar Publisher: O'Reilly 2015.

UE18CS314: Applied Cryptography (4-0-0-4-4)

Cryptography is the science of securing data by using mathematical concepts. Cryptography involves the authentication and verification of data in all domains by applying Cryptographic protocols.

Course Objectives:

- Enable to learn the fundamental concepts of cryptography and utilize these techniques in computing systems.
- Discuss about various encryption techniques.
- Understand the concept of public key Cryptography.
- Introduce message authentication and hash function.
- Provide Lab sessions for each unit to help gain deeper insight into Cryptography.

Course Outcomes:

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

- Classify the symmetric encryption techniques.
- Illustrate various Public key cryptographic techniques.
- Evaluate the authentication and hash algorithms.
- Discuss authentication applications.

Course Content:

Unit 1 : Classical Ciphers

Introduction to cryptography, cryptanalysis, and cryptology, Overview of cryptography, Basic Cryptographic primitives, Classical ciphers: substitution cipher – Caesar, Playfair and Hill cipher, Transposition cipher – Railfence, Columnar and Double columnar, Cryptanalysis of classical ciphers, Introduction to probability, Conditional probability, Law of Total probability, Shannon's theorem, One-time-pad encryption, Limitations of One-Time-Pad.

12 Hours

Unit 2 : Symmetric Key Cryptography

Introduction to symmetric key cryptography, Pseudo Random Numbers, Feistel Cipher, Sbox and E-box, Initial and Final permutations, Data Encryption Standard (DES), Cryptanalysi an avalanche effect, AES (Advanced Encryption Standard), Key Scheduling, Block and Stream ciphers.

12 Hours

Unit 3 : Public Key Cryptography

Introduction to Public key cryptography, Modes of operation, Prime number, Primitive root Modular arithmetic, Polynomials, Diffie Hellman Protocol(DH Protocol), Elgamal crypto systems, Prime Factorization, Rivest–Shamir–Adleman cryptosystem (RSA), Applications.

12 Hours

Unit 4 : Key management Hashing Techniques

Key management and Distribution Center(KDC), Birthday attack, Zero knowledge protocols, Message Digest algorithm (MD5), One-way function, Collision Resistant Hash Function (CRHF), Secure Hash Algorithm (SHA), Applications.

10 Hours

Unit 5 : Authentication using Cryptography

Identification protocols, Digital Signature (DS), Elliptic Curve cryptography-based signature (ECDSA), RSA based signature, Message Authentication Code (MAC), Cipher Block Chaining MAC (CBC-MAC), Different areas where cryptography needs to be applied.

10 Hours

Tools / Languages : Seed lab ,C-language.

Text Book:

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

Reference Book:

1: “Cryptography and Network Security”,Behrouz A Forouzan, Tata McGraw-Hill, 2007.

UE18CS315 : Database Technologies (4-0-0-4-4)

This course provides a systematic approach with an in-depth analysis of advanced database areas. This also covers new advanced database systems.

Course Objectives:

- Storage Techniques, Indexing Mechanisms.
- Query Processing and Query optimization techniques.
- Parallel and Distributed Databases.
- NoSQL Databases and Big Data systems.
- Design and build specialized database applications.

Course Outcomes:

At the end of this 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 the NoSQL (non-relational database) approach to the “Big-Data” problem.
- Apply techniques learnt and build specialized database applications.

Pre-Requisite:UE18CS252 Database Management System.

Course Content:

Unit 1 : Relational Data Model and Storage Formats and Indexing

Review of Relational Design theory – Functional dependencies, normalization, Overview of secondary storage, RAID and flash storage, Storing tables: row-wise, column database, database buffer, Indexing: concepts, clustered and non-clustered indices, B+-tree indices, multiple key access, hashed files, linear hash files, bitmap indices, Index definition in SQL, ++R-trees.

10 Hours

Unit 2 : Query Processing and Optimization

Memory Hierarchy, accelerating access to Secondary Storage, Disk Failures, Physical-query-plan operators – one and two pass algorithms, buffer management, Query Compiler – Algebraic laws for improving query plans, Parse tree to Logical Query plans, cost based Plan selection, choosing order for joins.

12 Hours

Unit 3 : Parallel and Distributed Databases

Avenues for parallelism: I/O parallelism, interquery, inter-query and intra operation parallelism, databases for multi-core machines. Parallel Algorithms on Relations, Map-Reduce Parallelism Framework, Distributed databases - Distributed data storage, distributed transactions, commit protocols, concurrency control in distributed databases, heterogeneous and cloud-based databases., Distributed query processing, Distributed Locking, Peer-to-Peer Distributed Search.

12 Hours

Unit 4 : NoSQL and Speciality Databases

Characteristics of NoSQL, Categories of NoSQL systems, NoSQL Databases -Document Databases with Example (MongoDB, CouchDB), Column Oriented Databases with Example (Cassandra), Key-Values Stores with Example (Riak,Voldemort), Graph Databases with Example (Neo4J). Specialty Databases: In-Memory Databases for RDBMS (VoltDB) and Key -Value Store (Redis).

12 Hours

Unit 5 : Design and Implementatio of Databases Systems

Current trends in database system design and Implementation of decision support, data warehousing and data mining applications including Big Data, support for data analytics and machine learning.

10 Hours

Tools/Languages: MySQL, Oracle.

Text Books:
1. “Database Systems: The Complete Book”, , H Garcia-Molina, JD Ullman and Widom, 2nd Ed., Pearson, 2013.
2. “Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement”, Eric Redmond and Jim R. Wilson, O'Reilly, 2012.

Reference Book(s):
1:“Database System Concepts, Silberschatz, Korth and Sudarshan, McGraw Hill, 6th Edition, 2013.
2: “Fundamentals of Database Systems”, Elmasri and Navathe, Pearson Education, 7th Edition, 2015.
3**:** “NoSQL Distilled”, Pramod J Sadalage and Martin Fowler, Addison Wesley, 2012.

UE18CS316 : Computer Graphics and Visualization (4-0-0-4-4)

Computer Graphics is a sub field of Computer Science which studies methods for digitally synthesizing and manipulating visual content. CG deals with generating images using 3D modeling, shaders, ray tracing, computer animation. CG is responsible for displaying art and image data effectively and meaningfully to the consumer.

Course Objectives:
- Impart the basics of computer graphics, different graphics hardware systems and applications of computer graphics.
- Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis using openGL.
- Introduce the use of geometric transformations on graphics objects and their application in composite form and its implementation.
- Impart frame extraction with different clipping algorithms and transformation to a graphics display device.
- Introduce projections and visible surface detection techniques for display of 3D scene on 2D screen and rendering of projected objects to naturalize the scene in 2D view.

Course Outcomes:
At the end of this course, the student will be able to:
- Demonstrate the fundamentals of computer graphics and display pipeline systems.
- Be able to draw different 2D objects using scan conversion algorithms and also fill basic objects and perform their comparative analysis.
- Use geometric transformations on graphics 2D objects and demonstrate their application in composite form.
- Be able to extract a 2D object using clipping algorithms and apply transformations to a graphics display system. Implementation of all the algorithms using openGL.
- Apply graphics in greater depth to more complex courses like Image Processing, Virtual and Augmented Reality, etc...

Pre-Requisite: UE18CS202- Data Structures.

Course Content:
Unit 1 : Introduction
Applications of Computer Graphics, A Graphics System: Images - Physical and Synthetic, Imaging Systems, The Synthetic Camera Model, The Programmer’s Interface, Graphics Architectures, Programmable Pipelines. Graphics Programming: Programming Two Dimensional Applications. The OpenGL: The OpenGL API, Primitives and Attributes, Color, Viewing, Control Functions, Polygons.

12 Hours

Unit 2 : Implementation
Basic Implementation Strategies, Four Major Tasks, Clipping, Line-Segment Clipping, Polygon Clipping, Clipping of Other Primitives, Clipping in Three Dimensions, Rasterization, Bresenham’s Algorithm, Polygon Rasterization, Hidden-Surface Removal, Anti- Aliasing.

12 Hours

Unit 3 : Geometric Objects and Transformations-I
Scalars, Points and Vectors, Three-Dimensional Primitives, Coordinate Systems and Frames, Modeling a Colored Cube, Overview of 2D Transformations: Rotation, Translation and Scaling, Affine transformations.

12 Hours

Unit 4 : Geometric Objects and Transformations-II
Transformation in Homogeneous Coordinates, Concatenation of Transformations, OpenGL Transformation Matrices, Interfaces to Three Dimensional Applications, Quaternion's.

12 Hours

Unit 5 : Rendering, Viewing and Animation
Classical and Computer Viewing, Viewing with a Computer, Positioning of the Camera, Simple Projections, Projections in OpenGL. Light & Shades: Light & Matter, Light Sources, Global Illumination.Introduction to Rendering and Animation.

10 Hours

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

Text Books:

1. "Interactive Computer Graphics - A top down approach wit WebGL",. Edward Angel and Dave Shreiner, Pearson Education , Indian Edition - Reg Office: Chennai, Seventh edition, 2016.
2. "OpenGL Programming Guide": Mason Woo, Jackie Neider, Tom Davis, Dave Shrenier: 3rd Edition, openGl version 1.2, Addision Wesley, 1999.

Reference Book:
1: "Interactive Computer Graphics: A Top-Down Approach with WebGL", Edward Angel, Pearson Education, 7th Edition,2015.

UE18CS321 : Principles of Programming Languages(4-0-0-4-4)

Principles of Programming Languages is a course to understand the various design choices made by Language designers and the philosophy behind these choices. The course is structured in a way that explores various programming paradigms and their features.

Course Objectives:

- Enable students to learn constructs in a language.
- Enable students to design a new construct/ language.
- Enable students to choose appropriate language for real world problem solving, based on the required features.
- Enable students to evaluate various language design features considering the programming paradigm.
- Introduce various paradigms and their support in language design.

Course Outcomes:

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

- Choose a particular language for problem solving depending on the application domain.
- Analyze and compare programming language concepts.
- Analyze the implementation issues related to a language design.
- Identify the language design features of any language and evaluate them.
- Apply various language features in solving real-world problem.

Course Content:

Unit 1 : Preliminary Concepts

Reasons for Studying, Concepts of Programming Languages,Programming Domains Language Evaluation Criteria, Influences on Language Design, Language Categories, Programming Paradigms – Imperative, Object Oriented, Functional Programming, Logic Programming, Programming Language Implementation – Compilation and Virtual Machines, Programming Environments. Names, Binding, Type Checking and Scopes: Names, Variables, Binding of Attributes to Variables, Type Bindings, Type Inferencing, Type Checking, Strong Typing. Case Study: Linux utilities and Program Debuggers for languages such as C, Python.

12 Hours

Unit 2 : Type Checking and Scopes

Type Equivalence, Scope, Scope andLifetime, Referencing Environments. Data types: Introduction, Primitives, Character, User Defined, Array, Associative, Record, Union, Pointer and Reference Types, Design and Implementation Issues Related to These Types, Names, Variable, Concept of Binding,Type Checking, Strong Typing, Type Compatibility, Named Constants, Variable Initialization. Expressions and Statements: Short Circuit Evaluation, Mixed Mode Assignment, Assignment Statements, Cascading Operators Case Study: Variable scopes and lifetimes in various languages such as C, Java and Python.

12 Hours

Unit 3 : Control Structures

Statement Level, Compound Statements, Selection, Iteration, Unconditional Statements, Guarded Commands. Subprograms and Blocks: Fundamentals of Subprograms, Scope and Lifetime of Variable, Static and Dynamic Scope, Design Issues of Subprograms and Operations, Local Referencing Environments, Parameter Passing Methods, Overloaded Subprograms, Generic Subprograms, Parameters that are Subprogram Names.

Case Study: Shallow binding and Deep binding of parameters in Function/procedure calls and their implications. Suitable languages can be chosen as demonstration medium.

12 Hours

Unit 4 : Functions

Functions: Design Issues for Functions, User Defined Overloaded Operators, Co- Routines and Function Closures. Abstract Data types: Abstractions and Encapsulation, Introduction to Data Abstraction, Design Issues, Object Oriented Concepts with Reference to Java and Python. Case Study: Object oriented concepts demonstration through suitable language. (Java/Python). Clarity on Immutable and Mutable Objects can be imparted with reference to these languages.

10 Hours

Unit 5 : Exception Handling

Exceptions, Specifications, Exception Propagation. Logic Programming Language: Introduction and Overview of Logic Programming, Basic Elements of Prolog, Application of Logic Programming. Functional Programming Languages: Introduction, Fundamentals of FPL, Applications of Functional Programming Languages and Exploration of the Features, Comparison of Functional and Imperative Languages.
Case Study: Logic Programming and Functional Programming feature demonstration with suitable languages such as Prolog/LISP and Haskell.

10 Hours

Tools / Languages : Various compilers and Debuggers as gcc, g++, Ada, Python, Ruby, Java, Prolog, Haskell, GDB, PDB**.**

Text Book:

1. "Concepts of Programming Languages", Robert W Sebesta, Pearson Education, 10thEdition, 2012.

Reference Book(s):

1: "Programming Language Pragmatics", Michael L Scott, Elsevier, 3rd Edition, 2009.
2. "Programming Languages Design and Implementation", Pratt and Zelkowitz, Prentice
Hall/ Pearson Education, 4th Edition, 2001.

UE18CS322 : 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.

Course Objectives:
- Provide an introduction to Big Data.
- Introduce computational and storage technologies for Big Data.
- Introduce algorithms for processing Big Data.
- Introduce programming tools, 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.
- Apply and differentiate between algorithms for processing Big Data and Normal Algorithms.
- Applying Big Data techniques in real life problems through a group based project.

Pre-Requisite: UE18CS202-Data Structures, UE18CS251- Design and Analysis of Algorithms.

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, Case Study: Google. YARN introduction.

12 Hours

Unit 2 : Big data infrastructures (Compute/Storage)
Overview of Hadoop Ecosystem, Introduction to sample Big Data Algorithms – matrix multiplication and pagerank Relational operators on Map-reduce, case study: HIVE, Other storage – Hbase/Cassandra.

12 Hours

Unit 3 : In memory computation
Issues with Hadoop, Spark and Scala/PySpark programming model, Transformations and Actions, Spark SQL, Spark architecture – RDD, DataFrames, Wide and Narrow dependencies, Complexity of Big Data algorithms – Communication Cost complexity model.

12 Hours

Unit 4 : Streaming analysis Streaming analytics use cases, Streaming Spark, Kafka – use cases, architecture, Streaming Algorithms –sampling, set membership – Bloom Filters counting, Counting unique elements – Flajolet Martin Algorithm.

10 Hours

Unit 5 : Advanced Analytics on Big Data
Clustering algorithms - k means and Collaborative filtering, Scaling Neural Networks for Big Data, Case Study MLLib.

10 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.
3: “Hadoop: The Definitive Guide”, Tom White, O’Reilly, 4th edition, 2009.

UE18CS323: Graph 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 combinatorics and its application in problem solving.

Course Objectives:

- Familiarize with concepts and abstraction of Graph Theory.
- Up skill students with computer representation of graphs and algorithms.
- Introduce students with advanced concepts in graph theory and applications.
- Teach graph theoretical modeling of problems and solution.

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 inclusion and exclusion and generating function to solve problems.
- Solve Recurrence Relations of first and second order

Pre-Requisite:UE18CS151 – Problem Solving with C, UE18CS202 – Data Structures.

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– Combinational and geometric graphs, Planar graphs –Different representation of a planer graph.

12 Hours

Unit 2: Coloring, Covering and Partitioning

Chromatic number – Chromatic partitioning – Chromatic polynomial – Matching – Covering – Four Colour problem – Counting, Register Allocation using graph coloring.

12 Hours

Unit 3: Graph Applications

Shortest Path Problem, Finding Articulation Points, Reliable Communication Network Problem, Chinese Postman Problem, Optimal Assignment Problem, Time Table Problem, Graph Databases, Graphs for social network analysis

12 Hours

Unit 4: Inclusion, Exclusion, Generating Functions

The principle of inclusion and exclusion, Generalizations of the principle, derangements, Rook polynomials. Generating functions: Introductory examples, Definition and examples– calculational techniques, partitions of integers, The exponential generating function, The summation operator.

10 Hours

Unit 5: Recurrence Relations

First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients, The Non-homogeneous Recurrence Relation, Generating Functions for second order recurrence relations.

10 Hours

Tools / Languages : C- Language.

Text Book:

- "Graph Theory: With Application to Engineering and Computer Science", Narsingh Deo, Prentice Hallof India, 2017.

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: Latest Web based resources.

UE18CS324 : 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's important to understand why blockchain is different and how it works in comparison with technologies of the past.

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.
- Learn various vulnerabilities and security mechanisms of blockchain.

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.
- Identify the threats of blockchain and deploy security mechanisms.

Pre-Requisite:UE18CS202-Data Structure**s**

Course Content:
Unit 1 : Blockchain Introduction
Key Blockchain Concepts, Nodes, Cryptocurrency, tokens, Public Ledger, Peer to peer Network, Types of blockchain, Permissioned blockchain model, Permission-less blockchain model, Blockchain Construction.
10 Hours

Unit 2 : Cryptography
Machines that encrypted data in the past, Modern encryption, Private and public keys, Hash functions, From blocks to hashes, Hash Pointer, Markle tree, Ledgers, Transactions and trade, The public witness, Computers that witness, Distributed Consensus, Smart contract design, Bitcoin Blockchain Network.
12 Hours

Unit 3 : The structure of the network: consensus algorithm
Proof of Work, Proof of Stake, Delegated Proof of Stake, Proof of Authority, Proof of Elapsed Time, Proof of Capacity, Proof of Space, Proof of Burn, RAFT, PAXOS, Byzantine Fault Tolerance System, PBFT.
12 Hours

Unit 4 : Second generation applications of Blockchain technology
Smart contracts: origins and how they function, Creating and deploying smart contracts, Tokens, Token standards,Second generation tokens Decentralized applications, How are DApps constructed?, Decentralized Autonomous Organizations (DAOs), Hyperledger Fabric: Blockchain-as-a-service (BaaS), Architecture and core components, Hyperledger fabric model, Working on hyper ledger and transaction processing.
12 Hours

Unit 5 : Blockchain Security
Blockchain vulnerabilities, Smart contract vulnerabilities, Blockchain on CIA security triad, Blockchain based DNS security platform, deploying blockchain based DDOS protection.
10 Hours

Tools / Languages : Claynet, Python.

Text Book:
1. “Introduction to Blockchain Technology”, Tiana Laurence, 1st edition, Van Haren Publishing, 2019.

Reference Book(s):
1: “Hands-On Cybersecurity 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 (19 July 2016).

UE18CS325 : Web Technologies II (4-0-0-4-4)

Web Technologies II course introduces the single page applications,demonstrates the usage of MEAN technologies to build rich web applications and securing them from various attacks.

Course Objectives:

- The AJAX concepts and techniques.
- Various patterns of AJAX communications and Reverse AJAX Techniques.
- NodeJS and web services through ExpressJS.
- The front end framework Angular 9.
- The non-functional side of the WWW.

Course Outcomes:

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

- Design Advanced Web Applications, with techniques like AJAX.
- Understand the patterns of UI update and Server-Push techniques.
- Use Node.js as a Server-Side Framework to develop web sites that provide fast and reliable content as well use ExpressJS Middleware.
- Build a rich Web application using the most recent Angular 9.
- Develop robust web sites that are immune to malicious web attacks.

Pre-Requisite: UE18CS204 – Web Technologies I.

Course Content:

Unit 1 : AJAX

JS objects, prototype inheritance, Ajax - Hidden Frames Technique, Image-Based AJAX, XMLHttpRequest, JSON and XML, Fetching Binary Data with XHR, Cross-Domain Access (CORS), Maintaining History in AJAX Calls.

10 Hours

Unit 2 : AJAX Patterns and Reverse Ajax

Predictive Fetch, Multi-Stage download, Periodic Refresh and Fallback Patterns, Submission Throttling, Reverse Ajax concepts: HTTP Streaming and Long Poll, HTML5 Server Sent Events,Service Oriented Architecture Concepts (Principles of REST, Architecture of SOAP-Based Services, REST vs. SOAP Based Services).

12 Hours

Unit 3 : NodeJS and Express JS

Introduction to Node JS, HTTP methods and Verbs, NodeJS process and child process, buffers, streams, File system, timers, events, call backs, query string, TLS/SSL and web module, Leveraging with Express REST API's, Express Installation and Server setup, Building the application stack, Routing, List API, Create API, Error Handling, Express Middleware, Server Setup, Express Scaffolding and Templates, Cookies & File Upload.

12 Hours

Unit 4 : Typescript & Angular 9

Typescript basics , Introduction to Angular, Modules, Components, Component Lifecycle, Angular forms, controls and Validations, Template and Views, Component metadata, Data binding, Directives, Pipes, services and dependency injection.

12 Hours

Unit 5 : Non-Functional Aspects of the Web:

Performance Considerations - Timeouts, Retries, Handling Server Errors, Multiple Requests, The HTTP 1.1 Two Connection Limit, Caching on the Client Side, Compression of Data, HTTP 2.0 – New Features, HTTP 2.0 vs. 1.1. Security – Web Attack Surfaces, Reconnaissance Review, Various Vulnerabilities and Precautions, SQL Injection, XSS, CSRF.

10 Hours

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

Text Books:

1. “Professional AJAX”, Nicholas C. Zakas et. al, Wiley Publishing, 2nd Edition, 2007.
2. “Node.js, MongoDB and Angular Web Development: The definitive guide to using the MEAN stack to build Web Applications” ,Brad Dayley , Brendan Dayley , Caleb Dayley, Addison-Wesley Professional publishing,2nd Edition, 11 October 2017.

Reference Book(s):

1: "Web Application Security, A Beginner's Guide" ,Bryan Sullivanand Vincent Liu, McGraw Hill Education; 1 edition (10 January 2012).

UE18CS351 : 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 toolchain 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.

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 behavior 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.

Pre-Requisite: UE18CS202- Data Structures, UE18CS254- Theory of Computation.

Course Content:

Unit 1 : Compilers

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.

10 Hours

Unit 2 : Syntax Analysis

The role of the parser, Syntax Error Handling, Error-Recovery Strategies. **Top-down parsing:** Recursive Descent Parser(RDP) with Backtracking**,** LL(1) Parser. **Bottom-up parsing :** Shift-Reduce Parsing, LR(0), SLR, viable prefixes, CLR, LALR.

12 Hours

Unit 3 : Syntax-Directed Translation
Syntax-directed definitions, Evaluation orders for SDD's, Applications of Syntax-Directed Translation, Syntax-directed Translation Schemes – Postfix Translation Schemes. **Parser Stack Implementation:** 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.

12 Hours

Unit 4 : Intermediate-Code Generation
Variants of Syntax Trees – Directed Acyclic Graphs for Expressions, Three-Address Code – Addresses and Instructions, Quadruples, Triples, Indirect Triples, SSA Form, Control Flow Graph. Machine Independent Optimization: Different Optimizations, Optimization of Basic Blocks. Data Flow Analysis : Live-variable analysis, Next-use algorithm.

12 Hours

Unit 5 : Run-Time Environments
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.

10 Hours

Tools/Languages : Lex and Yaac.

Text Book:

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.

UE18CS352 : Cloud Computing (4-0-0-4-4)

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.

Course Objectives:
- Introduce the rationale behind the cloud computing revolution and the business drivers
- Introduce various models of cloud computing.
- Introduce differences between traditional monolithic applications and cloud native application architectures
- Introduction on how to design cloud native applications, the necessary tools and the design tradeoffs.
- Introduce and design distributed systems for scalability.
- 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.
- Build and deploy applications for the cloud and understand the security implications.
- Apply the fundamentals of distributed systems design to cloud computing.
- Demonstrate design tradeoffs while designing cloud applications.

Pre-Requisite: UE18CS301- Computer Networks, UE18CS302- 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.
12 Hours
Unit 2 : Virtualization
Hypervisor - Types, Paravirtualization 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.
12 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.
12 Hours
Unit 4 : Cloud Controller
Master-slave v/s p2p models, Resource allocation, Scheduling algorithms, Cluster coordination – consensus, Fault Tolerance - faults and partial failures, Unreliable communication, Cluster coordination - leader election, distributed locking, Case Study: Zookeeper - distributed consensus infrastructure.
10 Hours
Unit 5 : Performance, Scalability and Security in Cloud
Scaling computation - reverse proxies, Scaling computation - hybrid cloud and cloud bursting, Multitenancy, Multitenant databases, Failure detection - checkpointing and application recovery, Cloud security requirements - physical/virtual security, Risk management, security design patterns, Security architecture,

legal and regulatory issues, Authentication in the cloud: Keystone, Cloud Threats – DoS, Economic Denial of Sustainability.

10 Hours

Tools / Languages : Amazon AWS (or equivalent), Docker, Kubernetes, github, NoSQL databases, Flask.

Text Book(s):

1. "Distributed and Cloud Computing", Kai Hwang, Jack Dongarra, Geoffrey Fox.ISBN: 978-0-12-385880-1, Morgan Kaufmann, 2012.
2. "Moving to the clouds: Developing Apps in the new world of cloud computing", Dinkar Sitaram and Geetha Manjunath. Syngress, 2011.
3. "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.

UE18CS353 : Object Oriented Analysis and Design with Software Engineering(4-0-0-4-4)

This course deals with the Software development life cycles, their individual phases, principles, methods, procedures and tools associated along with a flavor of Object Oriented Methodologies and designs. This also deals with making of choices, implications of making choices and exposes students to the OO and Software Eco-system which will be experienced by students in an post college environment. The theory is supplemented with a project which provides the hands on experiences of working with teams.

Course Objectives:

- Ensure the relevance and need of an engineering approach to software development.
- Learn Software Engineering concepts.
- Ensure exposure and appreciation of Object Orientation in design.
- Expose students to the tools available as part of the Software Development and Product Development Life Cycle.
- Enable the students to practice the principles of Software Product Development.

Course Outcomes:

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

- Relate to the challenges of Software Development and relate to SoftwareEngineering as a methodical approach for development.
- Use Software Development Life Cycles with an understanding of when and where to use.
- Use Object Oriented design approach with an understanding of how to use it.
- Work through and produce different artifacts expected at each phase of the lifecycle.
- Work on a project plan, track and manage projects and appreciate the importance and usage of quality and metrics in Software Development.

Course Content:

Unit 1 : Introduction, Requirements Engineering and Project Management

Understandthe context of Software Engineering; Fundamental drivers of Software Engineering.Generic Process framework, Phases in the development of software, Phases in maintenance of a product, Product life cycle Phases.Software development models including waterfall model, Incremental model, Evolutionary model, Agile SCRUM model, CBSE, SoA.**Requirements Engineering** and verification, Requirements specification, management and traceability. **Software Project andProduct Management:** Planning a software development project with overview ofdifferent aspects of SE management and Estimation. Exposure tools like MS Project.

12 Hours

Unit 2 : Architecture and Design

Software Architecture, architectural drivers, architectural, choices and impacts, Introduction to architectural views, architectural styles and architecture/design patterns.Classical and Object Oriented system design and its techniques, Object Oriented Modeling as a design technique, Introduction to UML, Structural model and interaction models - class model, use case model, sequence model, state models, deployment model. Exposure to a design tool.

12 Hours

Unit 3 : Development and Implementation

System conception and class design with ATM as an illustration and GRASP, SOLID principles **Implementation:** Coding standards & guidelines, code Review/Peer Review. **Change, Build& Release Management**: Elements of a Configuration Management Systems, Baselines, Repository, the SCM process, Configuration Management Plan, Management of code versions, release versions, Patching and patch management.Exposure to code management tools like GitHub, Build Bot.

12 Hours

Unit 4 : Software Testing, Quality and Ethics

Software Testing and the Software Life cycle, Testing Strategies, Verification and Validation, Planning and Documentation, Manual test Techniques, Coverage Based Test Techniques, Fault and Documentation, Manual test Techniques, Coverage Based Test Techniques, Fault based test techniques, Error Based Test

Techniques, Comparison of Test Techniques, Test Stages, and Estimating Software Reliability. Exposure to a test tool Software Quality: Managing Software Quality, Ataxonomy of Quality attributes, perspectives on quality, The quality system, Software Quality assurance, The Capability Maturity Model,Software Metrics.Ethics in Software Engineering, Software Engineering v/s Hacking, Software Engineering in a Global Environment. Project/Assignment Reviews and presentation by teams.

10 Hours

Unit 5 : IT Services Management and Dev Ops

Introduction to **ITSM&ITIL**.Introduction to **DevOps,**stitching it all together, Pillars of DevOps – Collaboration, Affinity, Tools andScaling, Continuous Build, Integration, delivery, Automated Build, testing anddeployment. Exposure to a tool like Jenkins. Project/Assignment Reviews and presentation by teams.

10 Hours

Tools / Languages : Github, MS Project, Jupiter , Star UML/ Java.

Text Book:

1. "Software Engineering: Principles and Practice", Hans van Vliet 3rd edition Wiley India 2010.

Reference Book(s):

1:Object Oriented Modeling and Design with UML by Micheal Blalh and James Rumbaugh 2nd edition Pearson 2013.

2:Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling by Jennifer Davis, Ryn Daniels 3rd edition O'Reilly 2013.

3: Software Engineering, Ian Somerville, 9th edition 2009.

4:IEEE SWEBOK and Other Sources from Internet.

UE18CS354 : Cloud Computing Laboratory (0-0-2-1-1)

Course Objectives:
- Introduce working with a public cloud and the terminology associated with cloud services.
- Introduce different communication mechanisms.
- Introduce cloud native programming models.
- Introduce deployment tools on the cloud like Docker and Kubernetes.

Course Outcomes:
At the end of this course, the student will be able to:
- Work with a public cloud and work on.
- Build and deploy a sample applithe cloud.
- Demonstrate the use of tools in building cloud applications.
- Demonstrate their learnings through practical hands-on assignments.

Course Content:
1. Connecting to Public cloud and creating a VM on the public cloud; setup the firewalls to allow connection to the VM. Auto shutdown.
2. Setup a web server on the VM in the public cloud. Create a web page on the web server and access it from your desktop. Monitor cloud usage.
3. Introduction to REST apis.
4. Introduction to message queues for communication.
5. Docker images; deploying docker containers.
6. Dockerizing the micro service pap, network setup.
7. Load balancers on the cloud.
8. Introduction to kubernetes setup and sample deployment
9. Storing data persistently.
10. Zookeeper.

Reference Book(s):
1. “Docker in Action”, Jeff Nickoloff, Manning Publications, 2017.
2. “Cloud Native DevOps with Kubernetes”, John Arundel and Justin Domingus, OReilly, 2019.
3. Laboratory Manual prepared by the Department of Computer Science and Engineering, PES University.

UE18CS355 : Object Oriented Analysis and Design with Software Engineering Laboratory (0-0-2-1-1)

This course focuses on designing an application using Object Oriented Approach and Software Engineering Skills.

Course Objectives:

- Ensure the relevance and need of an engineering approach to Software Development.
- Learn Software Engineering concepts.
- Expose students to the tools available as part of the Software Development and Product Development Life Cycle.
- Enable the students to practice the principles of Software Product Development.

Course Outcomes:

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

- Relate to the challenges of Software Development and relate to Software Engineering as a methodical approach for development.
- Use Software Development Life Cycles with an understanding of when and where to use.
- Work through and produce different artifacts expected at each phase of the lifecycle.
- Work on a project plan, track and manage projects.
- Appreciate the importance and usage of quality and metrics in Software Development.

Course Content:

- Identify a project title and form team.
- Formulate requirements and use analysis tools to capture the detail requirement.
- Formulate requirements and use analysis tools to capture the detail requirement.
- Use Design tools to come up with high level and low level design.
- Use Design tools to come up with high level and low level design.
- Implementation of the design.
- Implementation of the design.
- Implementation of the design.
- Implementation of the design.
- Testing the developed application (manual and automated).
- Testing the developed application (manual and automated).
- Create installation packages using Devop tool.

.

Tools / Languages : Github, Jupiter, MS project, Start UML / Any OO Languages.

Reference Book(s):

1: Laboratory Manual prepared by the Department of Computer Science and Engineering, PES University.

UE18CS331: Generic Programming(4-0-0-4-4)

Generic programming is one of the most popular paradigms of programming. This provides a type agnostic, flexible way of developing data structures and algorithms. This also provides programming at compile time in compile time artefacts.

Course Objectives:

- Understand rationale behind generic programming - appreciate generic functions.
- Understand rationale behind generic classes.
- Understand and appreciate STL philosophy.
- Understand and appreciate programming at compile time.
- Understand and appreciate generics in Java and C#.

Course Outcomes:

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

- Develop programs using generic functions.
- Develop programs using generic classes.
- Develop programs using STL.
- Develop programs using compile time artefacts.
- Develop simple programs using Java and C# generics.

Pre-requisites:UE18CS151- Problem solving with C, UE18CS202- Data Structures, UE18CS251- Design and Analysis of Algorithms.

Course Content:

Unit 1 : Template Functions

Definition, Instantiation - Implicit and Explicit, Specialization,Type and Non-Type Template Parameter.

12 Hours

Unit 2 : Template Class

Instantiation, Templates and Static Members, Templates andInheritance, Templates and Composition, Templates and Friends, Template Member Functions, Dependent Type, Default Template Parameter, Nested Templates.

12 Hours

Unit 3 : STL

STL Philosophy, Efficiency of Algorithms, Separation of Behaviour from Container Classes, Functor and Iterator, Iterator Hierarchy, Adaptors, Examples of Containers and Algorithms, Traits and Policies.

12 Hours

Unit 4: Template Meta - Programming Overview

Compile-Time Programming Nature andLimitations of Template Meta-Programming,
Values, Functions, Branching, Recursion, Compile-Time "If"Conventions for "Structured" Template Meta Programming.

10 Hours

Unit 5: Generics in Java and C#
Generic Methods, Constructors, Type Inference, Bounded TypeParameters, Subtyping, Wildcards, Type Erasure, Overview of Generic Collection Classes, Generics in C#, Generic Constraints, Generics and Casting, Inheritance and Generics, Generic Methods, Generic Delegates, Generics and Reflection.

10 Hours

Tools / Languages : C++, Java, C#.

Text Book :

1. “STL Tutorial and Reference”, Musser, Derge and Saini, 2nd Edition, Addison-Wesley, 2001.

Reference Book(s):
1: “C++ Primer”, Lippman, Addison-Wesley, 2013.
2: “A tour of C++”, Bjarne Stroustrup, Addison-Wesley, 2013.
3: “Templates: The Complete Guide”, David Vandevoorde, Nicolai M Josuttis, Addison-Wesley, 2002.
4: “Java Tutorials”, Online Reference Link - https://docs.oracle.com/javase/tutor.
5: MSDN for C# generics.

UE18CS332 : Algorithms for Intelligence Web and 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.

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 the recommendation algorithms and Clustering Algorithms and their working.
- Understand the different web Applications.

Course Outcomes:

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

- Implement an efficient index for a document collection.
- Perform searches on a document collection, rank and evaluate results.
- Apply Machine Learning techniques in Information Retrieval Systems and Web Applications.
- Implement different recommendation algorithms and clustering algorithms.
- Design different intelligent web applications.

Pre-Requisite: UE18CS251 - Design and Analysis of Algorithms.

Course Content:

Unit 1 : Introduction to Information Retrieval and Web Scale Computing

Background, Architecture and Strategies of Information Retrieval (IR) Systems, IR Models, Boolean andExtended Boolean Models, Dictionary, Vocabulary, Positional Postings, Phrase Queries and Tolerant Retrieval. Introduction to Web and Intelligent Web Applications. Examples of Intelligent Web Applications.

10 Hours

Unit 2 : Indexing and Vector Space Model, Evaluation of IR

Algorithms for Indexing and IndexCompression, Vector Space Model for Scoring, tf-idf and Variants, Efficient Scoring and Ranking, Performance Measurement, Relevance Feedback, Query Expansion, Other IR Models.

10 Hours

Unit 3 : Web Applications and Search Algorithms

Web Search Basics, Economic Model of Web Search, Lucene as a Search Engine, Other search engines like Solr, Everything, Google. 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.

12 Hours

Unit 4 : Recommendation Algorithms

Instance based Learning, Introduction to Recommender Systems- Goals, Basic Models, Domain Specific Challenges in Recommender Systems, Neighborhood based collaborative filtering, Model based collaborative filtering, Content based Recommender System, Constraint based recommender systems. Analysis of the Paper "Cutting Edge Collaborative Recommender Algorithms "by Balazs Hidasi (from the book Collaborative Recommendations, Algorithms Practical Challenges and Applications).

12 Hours

Unit 5 : Design of Intelligent Web Application

Design of an Intelligent Web Application, User Requirements, Selecting Algorithms, Data Design, Design for Performance,Architecture of an Intelligent Web Application, Implementation Issues, Summary and Conclusion. Analysis of 3 Research papers (from Artificial Intelligence: Methods in Intelligent Algorithms Proceedings of 8th Computer Science On-line Conference 2019, Vol. 2. WI 2020: Web Intelligence, WI-IAT 2020, the 19TH IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Melbourne, Australia). Analysis of the paper "Context Aware Recommendations "by Yong

Zheng and Bamshad Mobasher (from the book Collaborative Recommendations , Algorithms Practical Challenges and Applications).

12 Hours

Tools/Languages : Scikit, Tensor flow, Solr, Lucene search engines, Python.

Text Book:

1. “Introduction to Information Retrieval”, Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze, ISBN: 9781107666399, Cambridge University Press, 2009.

Reference Book(s):

1: “Algorithms of the Intelligent Web”, Haralambos Marmanis, Dmitry Babenko,Manning Publishers, 2011.
2: “Recommender Systems – The Text Book “, Charu C. Agarrwal, ISBN- 978-3-319-29657-9, Springer 2016**.**
3: “Collaborative Recommendations, Algorithms Practical Challenges and Applications”, Shlomo Berkovsky, Ivan Cantador, Domonkos Tikk, ISBN – 978-981-3275-34-8, World Scientific 2019.

UE18CS333 : Digital Image Processing (4-0-0-4-4)

Digital 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 image fundamentals (acquisition, storage and viewing).
- Gain an insight to the mathematical transforms necessary for processing of grayscale and binary images .
- Assess the quality of an image and study the application of enhancement techniques in the spatial and frequency domains .
- Understand some types of noise that affects images and study techniques for denoising and restoration.
- Understand some of the techniques used for processing of 3D/ colour images.

Course Outcomes:

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

- Understand image formation and the role human visual system plays in perception of gray and colour image data.
- Apply image processing techniques in both the spatial and frequency (Fourier) domains.
- Assess the quality of an image and apply appropriate enhancement techniques.
- Design and evaluate methodologies for image segmentation.
- Conduct an independent study and analysis of feature extraction and image classification techniques.

Pre-Requisite: UE18CS251 – Design and Analysis of Algorithms.

Course Content:

Unit 1 : Introduction

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.

12 Hours

Unit 2 : Image Enhancement in the Spatial Domain

Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters, and Combining Spatial Enhancement Methods. **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: Slant Transform, Haar Transform and KL Transform.

12 Hours

Unit 3 : Morphological Image Processing and Segmentation: Preliminaries

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.

12 Hours

Unit 4 : Color Fundamentals and Basics of Compression

Color Models, Pseudocolor Image Processing, Basics of Full-Color Image Processing, Color Transformations, Smoothing and Sharpening, Color Segmentation, Noise in Color Images**.** **Image Compression**: Fundamentals - Image Compression Models, Some encoding techniques**.**

10 Hours

Unit 5 : Feature extraction

Scale Image Feature Transform, Image Pattern Classification: Patterns and Pattern Classes, Pattern Classification by prototype matching, Bayes Classifier for Gaussian Pattern Classes, Neural Networks and Deep Learning, Deep Convolutional Neural Networks.

10 Hours

Tools / Languages : Matlab

Text Book:

1. “Digital Image Processing”, Rafael C Gonzalez and Richard E. WoodsPrentice 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.

UE18CS334 : 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 meaning of text by various types of learning models including the recent approaches using deep learning and the significance of NLP pipeline in that meaning disambiguation process. The course also discusses disambiguation of syntax as a step of meaning disambiguation process.

Course Objectives:

- Learn the central themes, learning problem and the problem solving approaches used in NLP.
- Learn various learning models related to sequence labeling that is the basic building block in NLP.
- Learn how syntactic disambiguation is done in NLP.
- Learn how lexical and distributional semantics can be used for semantic disambiguation in NLP.
- Learn deep learning techniques and 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 central themes, central problem being solved in NLP and the learning approaches used in solving them.
- Get a grip on various sequence labeling approaches and applications in NLP.
- Get a grip on how syntactic ambiguity removal can contribute in overall disambiguation process.
- Apply comfortably and confidently appropriate branch of semantics depending on the problem being solved.
- Learn how to implement neural language model, NLP applications using neural techniques and utilize various transfer learning approaches in NLP.

Pre-Requisite: UE18CS303-Machine Intelligence.

Course Content:

Unit 1: Introduction

Introducing central problem and the three themes in Natural Language Processing: Types of ambiguity in natural language processing, Three themes in Natural Language Processing – Learning and Knowledge; Search and Learning; Relational, Compositional and Distributional Perspectives. **Text normalization:**Content and Function words, type vs. token, word tokenization and normalization, Morphological parsing of words – Porter stemmer, Lemmatization and Stemming, Sentence segmentation, n-grams. **Noisy Channel model:** Real world spelling error, Minimum edit distance algorithm, Concept of noisy Channel Model. **Learning models:** Linear classification, non-linear classification, learning without supervision.

10 Hours

Unit 2: Handling sequences of text

Language model n-gram language model, smoothing, discounting and back-off, Kneser-Ney smoothing, interpolation, perplexity as an evaluation measure. **Sequence labeling:** Sequence labeling as classification, sequence labeling as structure prediction, Viterbi algorithm and Hidden Markov Model, POS Tagging example, POS Tagging using discriminative models i.e. Maximum Entropy Markov Model (MEMM). Discriminative Sequence labeling with features-Conditional Random Field. **Other sequence labelling applications –** Named Entity Recognition: practical NER architectures, **Dialogue :** Sequence over utterances, chatbots – rule and corpus based.

12 Hours

Unit 3: Parsing - Disambiguating Structure

Syntactic parsing: Ambiguity presented by parse trees, CKY parsing, Chart parsing and Earley parser, 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. **Co-reference resolution:** Forms of referring expression, algorithms for coreference resolution – mention pair and mention ranking model, mention detection, classifiers using hand-built features. **Discourse:** Segmentation- topic and functional, relations, shallow discourse relation.

10 Hours

Unit 4: Semantics: Lexical and Vector Semantics

Word senses and relations between word senses, **WordNet**: A Database of Lexical Relations; **Word sense disambiguation** : supervised word sense disambiguation, **WSD** : dictionary and thesaurus methods, semi-supervised WSD, **unsupervised word sense induction Semantic relatedness based on thesaurus like WordNet** : Resnik similarity, Lin similarity, Jiang-Conrath distance, Extended Gloss overlap and Extended Lesk method. **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 and Glove, **Improving Word2vec**: FastText, Limitation of distributional methods.

12 Hours

Unit 5: Natural Language Generation and Neural Network Methods

Neural Sequence labelling - Recurrent Neural network language model for POS tagging. **Convolutional Neural Network for text:** word level and character level language model with CNN and Sentiment analysis. **Neural dialogue agents :** Seq2Seq Chatbots using encoder-decoder architecture, attention models. **Neural Question answering -** Information Retrieval based factoid question answering, knowledge based question answering, Neural Question answering. **Transfer learning in modern NLP -** BERT, ELMo, GPT, ULMfit.

12 Hours

Tools / Languages : Tensorflow, Spacy, NLTK,SCIKIT Learn, Python 3.x.

Text 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 github https://github.com/jacobeisenstein/gt-nlp class/blob/master/notes/eisenstein-nlp-notes.pdf.

Reference Book(s):

1: "Speech and Natural Language Processing", Daniel Jurafsky and James H. Martin, 2nd edition paperback,2013. The more up to date 3rd edition draft is available at http://web.stanford.edu/~jurafsky/slp3/.

UE18CS335 : Computer Network Security (4-0-0-4-4)

Give you overview and conceptual understanding of network related security aspects. Students will have opportunity well dwell in to technical "how to "with hands on sessions and with case – studies.

Course Objectives:

- Provide an overall view of what Computer and Network Security is all about and generate interest in this field to be able to take this as a further specialization area or a career path.
- Introduction of Perimeter Security (Firewall, IDS, IPSEC).
- Understand various access roles of an organization.
- To learn about how to maintain the Confidentiality, Integrity and Availability of a data.
- To understand various protocols for network security to protect against the threats in the networks.

Course Outcomes:

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

- Network Security facilitates protection of information that is shared between computers on the network.
- Perform various network attacks and their mitigation strategies.
- Helps in protecting personal data of clients existing on network.
- If there are various computers attached to a network, there may be some computers that may have greater access to information than others.
- Private networks can be provided protection from external attacks by closing them off from internet. Network Security makes them safe from virus attacks, etc.

Pre-Requisite: UE18CS301 – Computer Networks.

Course Content:

Unit 1: Introduction to Computer Network Security

Plagiarism, IOT, CIA, passive and active attack, Attack Surface Categories, Vulnerabilities, Threats and Attacks, Assets, Countermeasures, privacy, General Data Protection Regulation, security vs privacy, Data Breaches. Vulnerabilities by Category, Real Life Examples of Cyber Crime, IIoT Cyber-attacks, Ransomed medical devices, The Attack Landscape, MITM / Eavesdropping, Malware / Ransomware, Phishing, DOS, security framework, job outlook.

10 Hours

Unit 2: Network Security Analysis

Packet Sniffing & spoofing and TCP protocols : Packet Sniffing, Shared Networks, Packet Flow in the System, Promiscuous Mode, Monitor Mode, Packet Filter, Receiving Packets Using Raw Socket, Packet Sniffer, Pcap library. Types of Spoofing Attacks. SYN Flooding Attack, TCP Reset Attack, TCP Session Hijacking Attack.

12 Hours

Unit 3: Network Security Systems

Firewall, VPN using Firewall, IDS, IPS, Honeypot, snort, Building a Firewall using Netfilter, Kernel Modules, Testing our Firewall, Applications, Terminology, IP Tunnelling, IPSec. Intrusion Detection and Prevention, HIDS, NIDS, IT Security Management Overview.

12 Hours

Unit 4: Risk, DNS, Heartbleed

DNS Hierarchy, Zones, and Servers,DNS Query Process, Remote DNS Cache Poisoning Attack, Reply Forgery Attacks from Malicious DNS Servers, DNS Rebinding Attack, Protection Against DNS Cache Poisoning Attacks, Ddos. IT Security Management Overview, IT Security Controls, Plans, and Procedures, Fixing the Heartbleed Bug.

12 Hours

Unit 5: Cloud Security, Wireless Network Security

Cloud Computing Service Models and Layers, Security Issues in Cloud Computing. Bluetooth Security: Bluetooth Protocol Stack, Multiple Security Modes. Mobile Security: Security Concepts, Requirements, Architecture. Wireless Communications and 802.11 WLAN Standards Wireless Protected Access (WPA), IEEE 802.1x, 802.11i/ WPA2, Wireless Network Threats, ZigBee Security, Wireless Mesh Network Security. Giving hands on experience for relevant topics in the form of Lab or Assignment, Relevant cyber security sase for undergraduate students are discussed.

10 Hours

Tools / Languages : Seed labs, Wire shark, netwox tool, Scapy.

Text Book:

1. "Computer and Internet Security", Hands on Approach", Wenliang Du , 2nd Edition, 2019.

Reference Book(s):

1: "Computer Security: Principles and Practice", William Stallings, Lawrie Brown, Pearson, Indian Edition, 2010.

UE18CS336 : Wireless Network Communication (4-0-0-4-4)

Wireless Networks Communication is a dynamic field that has spurred tremendous excitement and technological advances. This course provides a comprehensive understanding of the fundamental principles, characteristics, performance limits of wireless systems, their security issues and the insights associated with their design.

Course Objectives:

- Introduce the emerging trends of wireless network technologies
- Compare and contrast the wireless network technologies depending on the usage models
- Explain the characteristics of wireless channels and analyze its impact during communication
- Discuss various design parameters of communication.
- Identify different attacks on wireless network and explore several mitigation approaches.

Course Outcomes:

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

- Identify and Apply the appropriate wireless technology for real time applications.
- Simulate the channel characteristics such as path loss, shadowing, analyze the wireless networks and understand the Multipath channel models.
- Analyze emerging enhancements such as Adaptive Modulation and Multiple Input Multiple Output System.
- Capture the transmitted packets of wireless networks and analyze them for the wireless communication protocols
- Determine the threats on wireless network and Apply wireless security mechanisms.

Pre-requisite Course: UE18CS301 – Computer Networks.

Unit 1: Overview of Wireless communication

Introduction, Wireless LAN Technology: IEEE 802.11, WPAN Technologies: Bluetooth, Zigbee, NFC, 6LOWPAN, LPWAN- LORA, Weightless, Wireless Local Loop (WLL)-LMDS, MMDS, WiMAX, Long Range Communication- Satellite Communication, Wireless communication analysis using Wireshark.

12 Hour

Unit 2: Overview of Wireless Communication-II

Cellular Network: Cellular System Fundamentals, Channel Reuse, SIR and User Capacity, Interference Reduction Techniques, Mobile Applications and Mobile IP, Tradeoff between Battery, Bandwidth and Distance, Wireless Channel Models: Path Loss and Shadowing Models, Millimeter Wave Propagation, Fading Models: Statistical Fading Models, Narrowband Fading, Wideband Fading Models.

12 Hours

Unit 3: Impact of Fading and ISI on Wireless Performance

Capacity of Wireless Channels, Digital Modulation and its Performance, Adaptive Modulation: Adaptive Transmission System, Adaptive Techniques; Variable-Rate Variable-Power MQAM, Multiple Input/ Multiple Output (MIMO): Narrowband MIMO Model, Parallel Decomposition of the MIMO Channel, MIMO channel capacity, MIMO Diversity Gain: Beamforming, Diversity/Multiplexing Tradeoffs, Space-Time Modulation, Frequency-Selective MIMO Channels, Smart Antennas.

12 hours

Unit 4: Multicarrier Systems

Data Transmission using Multiple Carriers, Multicarrier Modulation with Overlapping Subchannels, OFDM, OFDMA, Single- carrier FDMA, Challenges in Multicarrier Systems, Multiuser Channels: The Uplink and Downlink, Multiple Access, Random Access, Downlink (Broadcast) Channel Capacity, Uplink (Multiple Access) Channel Capacity, Uplink/Downlink Duality.

10 Hours

Unit 5: Security

Attacks on Wireless Network, Attacks on Wireless Clients, WEP-Wired Equivalent Privacy Protocol Security, WiFi Security: WiFi Protected Access, WiFi Protected Access 2, WPA2 Wireless Enterprise Network, RADIUS, Handling Rogue Access Points, Theory of Defense for security wireless Networks.

10 Hours

Tools/ Language: Wireshark, Claynet/packetracer.

Text Book

1. "Wireless Communication", Andrea Goldsmith, First Edition, Cambridge University Press, 2012

Reference Book(s):

1: "Wireless Communication Networks and Systems", by Cory Beard and William Stallings,1st edition, pearson, 2015.

2: "Wireless Network Security: A Beginner's Guide" by Tyler Wrightson, McGraw-Hill Education; edition, 2012.

UE18CS337 : Cyber Forensics (4-0-0-4-4)

Cyber Forensics course provides a deep understanding of the techniques to gather,protect and report the digital confirmations.

Course Objectives:
- The Cyber Security issues, the Digital Forensics process and the Hard disk structure.
- The process of Data Acquisition and the structure of FAT and NTFS file system on Windows operating system.
- The Structure of Linux File system (EXT3/EXT4) and the file carving process.
- The Android Mobile device forensics and Multimedia Steganography procedures.
- The procedure for Email Forensics Analysis and Final report writing as per the court of law.

Course Outcomes:
At the end of this course, the student will be able to:
- Understand the phases in Forensic Investigation process and make out the internal structure of HDD and booting process.
- Use SleuthKit Library and Make an image of the Evidence with various open source tools and gain knowledge on FAT and NTFS file systems.
- Analyse the Unix/Linux File systems with exercises and do file carving using open source tools.
- Perform the Mobile device forensics and do Steganalysis for Multimedia forensics.
- Do Email forensics and know how to write a good report to be submitted to the court of law.

Course Content:
Unit 1 : Introductionto Forensic Process
Introduction to computer forensics, Forensics Investigation Process, Forensic Protocol for Evidence Acquisition, Digital Evidences, Types of computer forensics, Challenges in computer forensics, Understanding the Hard disks and File systems-HDD, SSD, Physical structure and Logical Structure of Hard Disk, Tracks, Sector, Cluster, Disk Partitions and Boot process, Open source tools.

10 Hours

Unit 2 : Data Acquisition and Windows File system Forensic Analysis
Building a Forensics Work station with The Sleuth Kit, Case Study-Data Acquisition – Imaging using Access Data FTK Imager and Encase ,Recovering files from the images using Encase, Examining FAT File system, Examining NTFS File system, Case study - NTFS Timestamp Analysis, Autopsy Tool Hands-on.

12 Hours

Unit 3 : Linux File system Analysis andFile carving
Unix/Linux file systems (Ext2/Ext3), Unix/Linux Forensic Investigation: Unix/Linux forensics, investigation steps and technologies, Case Study: Memory Acquisition of Linux System using LiME , Principles of file carving, Header/Footer carving, Bitfragment Gap carving, Case Study- Image File Foremost File Carving tool.

10 Hours

Unit 4 : Android Mobile device Forensics and Multimedia Forensics
Mobile Device Forensic Investigation, Storage Location, Acquisition Methods, Data Analysis of Facebook, Whatsapp, Case study using Android Virtual Device, Steganography Techniques and Tools, Steganalysis Techniques and Tools, Case study-Steganalysis using OpenStego, Anti Forensics Practices-Data Wiping and Shredding, Trail Obfuscation, Encryption, Data Hiding, Case Study-Anti forensic detection using Stegdetect.

12 Hours

Unit 5 : Email Forensics and Investigative reports and Legal Acceptance
Email Forensics, Recovering emails, Email Header Analysis, Case Study-e-Discovery from Enron Corpus, reparation work for report Writing, Structure of the report, Characteristics of a good report, Document design and good writing practices, Legal Acceptance, Case Study – Legal Acceptance in Autopsy tool, Incident Response process.

12 Hours

Tools / Languages : Open source tools on Forensics.

Text Book:

1. “Introductory Computer Forensics-A Hands-on practical Approach”, by Xiaodong Lin, Springer, 2018.
2. “Practical Cyber Forensics- An Incident-Based Approach to Forensic Investigations”, by Niranjan Reddy, A Press, 2019.

Reference Book(s):

1: “Digital Forensics Workbook_-Hands-on Activities in Digital Forensics”, by Michael K Robinson ,CreateSpace Independent Publishing Platform, 2015.

UE18CS338 : Enterprise and Resource Planning(4-0-0-4-4)

ERP is an undergraduate engineering course that deals with Enterprise Resource Planning software application solution that is utilized in organizations of various industries across the world. ERP addresses the enterprise needs of an organization by tightly integrating the various functions of an organization using a process view of the organization. In this course important internal sub-systems are studied in addition to the study of the life cycle of ERP implementation in medium to large enterprises.

Course Objectives:

- To learn the strategic importance of Enterprise Resource Planning systems in industry.
- To learn the basics of ERP, costs and benefits and the modules of ERP.
- To learn about Change Management, BPR and BPM.
- To learn key selections criteria, issues & risks involved in ERP implementation.
- To be aware of ERP related technologies and commercial ERP software.

Course Outcomes:

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

- Identify typical functionality of ERP sub-systems.
- Apply strategies for Change Management, BPR and BPM.
- Apply criteria to select ERP Package and Consulting Partner.
- Systematically develop plans for an ERP Implementation project and.
- Identify critical success factors and associated risks.

Course Content:

Unit 1 : Introduction to Enterprise Resource Planning Systems

Overview, Definition of ERP, Evolution of ERP Systems, ERP System Benefits and Challenges, Enterprise Processes, Extended ERP, Major ERP Players – Product and Consulting Companies, ERP Implementations. **ERP Implementation Life Cycle:** Life Cycle of an ERP Implementation Project, Phases, Methodologies, Types of ERP Projects, Deployment Strategies. **Business Case and ROI for ERP System:** Benefits and Costs of an ERP Implementation, Cost-Benefit Analysis.

12 Hours.

Unit 2 : Change Management

Reasons People Resist Change, Change Management Strategies, Organization Design, Change Management Team and Roles, Change Management Activities. Business Process Re-Engineering: Principles and Need, Definition, Phases, Pros and Cons, Keys to Success, Reasons for Failure, BPR Team and Roles, Process Selection and Diagnosis, Process Redesign, BPR and ERP, Benchmarking, Best Practices. Business Process Modelling and Business Modelling: BPM Need, Guidelines, As-Is and To-Be Modelling, Business Process Hierarchy, Standards for BPM, Process Modelling Software, Business Modelling, Integrated Data Modelling.

12 Hours

Unit 3 : ERP Functional Modules

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

12 Hours

Unit 4 : Procurement and Inventory Management

Procurement Process, Types, KPIs. Inventory Management Process, Types, Models, KPIs. **ERP Package Selection:** Selection Team, Selection Criteria, Parameters for Package Selection, Request for Proposal (RFP), Gap Analysis, ERP Market. **ERP Consulting Partner Selection:** Selection Criteria, RFP Process, In-house and Offshore Implementations, Pros and Cons. **Managing an ERP Project:** Scoping, Plan, Charter, Risk Management. Project Teams. Success or Failure of an ERP Implementation.

10 hours

Unit 5 : ERP and Enterprise Applications

Supply Chain Management (SCM), Customer Relationship Management (CRM), Product Life Cycle Management (PLM), Data Warehousing, Business Intelligence (DW-BI), ERP on Cloud, ERP for Manufacturing and Service Industries. Articles and Case Studies.

10 hours

Text Book:

1. "Enterprise Resource Planning- Text & Cases", Rajesh Ray, McGraw Hill Education, New Delhi, 2017.

Reference Book(s):

1: "ERP Demystified", Alexis Leon, McGraw Hill Education, 3rd Edition, 2014.
2: "Enterprise Resource Planning: A Managerial Perspective", Veena Bansal, Pearson Education India, 2013.

UE18CS341:Design Patterns (4-0-0-4-4)

Software quality attributes can be classified as internal and external. Code reusability, maintainability, and portability are the examples of internal software quality attributes. System performance, scalability, and availability are the examples of external software qualityattributes. Design patterns in general describe the best practices for a specific type of design problem based on the fundamental design principles. Low-level software design patterns deal with the internal software quality attributes. High-level architectural design patterns deal with the external software quality attributes. This course teaches students how to design quality software systems by applying both the software and architectural design patterns. In this course, application of design patterns is based on object-oriented methodology by using JAVA.

Course Objectives:

- Design principles beyond coding.
- Good habits in design.
- Appreciation for what to do and what not to do.
- Alternate design solutions.
- The intricacies of design.

Course Outcomes:

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

- Separate the interface from implementation in any complex problem.
- Identify the contexts where design patterns can be applied.
- Identify where not to apply design patterns.
- Recognize negative consequences of applying patterns using Anti-Patterns.
- Reliably refactor a large piece of software.

Pre-Requisite:UE18CS251 – Design and Analysis of Algorithms.

Course Content:

Unit 1 : Design Principle

Interface and Implementation, Open Closed Principle, LiskovSubstitution Principle, Dependency Inversion Principle, Integration Segregation Principle. What is a design pattern? describing design patterns, the catalog of designpattern, organizing the catalog, how design patterns solve design problems, how to select a design pattern, how to use a design pattern.

12 Hours

Unit 2 : Design Pattern Catalog

Structural patterns, Adapter, bridge, composite, decorator, facade, flyweight, proxy.

12 Hours

Unit 3 : Behavioral Patterns

Chain of Responsibility, Command, Interpreter, Iterator, Mediator,Memento, Observer, State, Template Method.

10 Hours

Unit 4 : Interactive systems and the MVC architecture

Introduction, The MVC architectural pattern, analyzing a simple drawing program, designin the system, designing of the subsystems, getting into implementation, implementing undo operation, drawing incomplete items, adding a new feature, pattern-based solutions.

12 Hours

Unit 5 : Introduction to Anti-patterns and Refactoring

What are anti-patterns? , Software Development anti-patterns, Software Architecture Anti-patterns, project management anti-patterns , Refactoring.

10 Hours

Tools / Languages : UML design tools , Python.

Text Book:

1. “Design Patterns” Erich Gamma, Richard Helan, Ralph Johman, John Vlissides, Pearson Publication,2013**.**

Reference Book(s):

1:“Design Principles and Design Patterns”, Robert C Martin, 2000.
2: “Object- oriented analysis, design and implementation” , Brahma Dathan, Sarnath Rammath, Universities Press,2013
3. “AntiPatterns **-** The Survival Guide to Software Development Processes”, Alexander Shvets, Online Reference at http://bit.ly/2e4nxzd.

UE18CS342 : 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 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 & challenges in Parallel Programming using popular frameworks.
- Familiarize with opportunities & challenges in Parallel Programming using popular languages.

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.
- Understand techniques for Parallelism Bugs Resolution.
- Program using popular parallel programming frameworks and languages trending industry.
- Engineer High performance migration of varied applications appreciating & assimilating varieties of parallelism.

Pre-Requisite:UE18CS151- Problem soving with C, UE18CS253 – Microprocessor and Computer Architecture.

Course Content:
Unit 1 : Fine Grained Parallelism
Review of Laws of Parallelism and Performance, Instruction Level Parallelism and Enhancement Techniques, Dependence Analysis, Prediction and Speculation, Vectorization and Pradication, Code Optimization, Cache optimized Programming.

12 Hours

Unit 2 : Coarse Grained Parallelism
Data, Task and Pipeline Parallelism, Pthreads, Multithreaded and Multi-Core architectures, GPUs and GPGPUs, Many-Core Heterogeneous Architectures.

12 Hours

Unit 3 : Parallelism Bugs and Resolution
Program Analysis, Memory Consistency Models, Data Races and Atomicity Violations, Deadlocks & Livelocks, Lock Free Data Structures.

10 Hours

Unit 4 : Parallel Programming Frameworks
OpenMP, Cuda, OpenCL.

12 Hours

Unit 5 : Parallel Programming Languages
UPC, Chapel, Concurrency in Mainstream Languages.

10 Hours

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

Text Book:

1. "Programming Massively Parallel Processors", David Kirk and Wen-mei Hwu ,3rd ,Morgan Kaufmann ,2016 .

Reference Book(s):

1: “Computer Architecture: A Quantitative Approach: John Hennessy David Patterson” ,6th Edition ,Morgan Kaufmann ,2017.
2: “Computer Systems: A Programmer's Perspective”, Randal E. Bryant, David R. O' Hallaron ,2nd ,Pearson ,2016.
3: Latest Web Resources and Papers .

UE18CS343 : 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.

Course Objectives:

- To impart hands-on knowledge on Advanced Machine Learning Topics.
- Introduce students to programming with TensorFlow and Keras tools.
- Provide in-depth coverage of Support Vector Machines.
- Introduce students to Deep Learning techniques – CNN and RNN.
- Introduce students to Reinforcement Learning and Generative Adversarial Networks.

Course Outcomes:

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

- Implement Machine Learning techniques with TensorFlow and Keras and develop simple game engines using Reinforcement Learning.
- Solve time-series related problems with RNN.
- Classify real-world data using Support Vector Machines.
- Classify images using CNN.
- Generate data in the form of images using GAN.

Pre-Requisite: UE18CS303 – Machine Intelligence.

Course Content:

Unit 1 : TensorFlow and Keras and Reinforcement Learning

Brief overview of DeepLearning Frameworks. **TensorFlow**: Installation, Creating and Managing Graphs,Lifecycle of a Node Value, Linear Regression, Gradient Descent, Visualizing Graphs using TensorBoard. **Keras**: Installation, Loading Data, Defining and Compiling Models, Fitting and Evaluating Models, Simple Neural Networks' Implementation,Fine-Tuning Hyper parameters. **Reinforcement Learning**: Learning to Optimize Rewards, Credit Assignment Problem, Temporal Difference Learning and Q Learning. **Case Study:** Learning to play a simple game using deep Q-learning -implementation.

10 Hours

Unit 2 : Support Vector Machines

A Very Brief Recap of the Support Vector Machine (SVM) Problem, Soft-Margin SVM (Noisy Data), Kernel Functions – Linear,Polynomial, Gaussian, Other Types, the SMO Algorithm, Multi-Class SVMs, Text-Classification, Building Applications.

12 Hours

Unit 3 : Recurrent Neural Networks (RNN) and Unsupervised Feature Learning

Recurrent Neurons, Memory Cells, Static and Dynamic Unrolling through Time, Variable-Length Input-Output Sequences, Training RNNs – Sequence Classifier, Predicting Time Series, Deep RNNs, LSTM Cell and GRU Cell, Text Classification with RNN, RNN Vs Naive Bayes, Seq2Seq with Attention , Bahdanau attention,Transformer Attention, Unsupervised Feature Learning – Autoencoders and Variational Auto Encoders.

12 Hours

Unit 4 : CNN, GAN and Transfer Learning

CNN: Architecture of CNNs, Filters, FeatureMaps, Max-Pool Layers, Other Pooling Types, Case Study: Image Recognition UsingCNN – Hands-On Implementation Using Keras. **Capsule Networks** – Introduction to Capsules, Dynamic Routing and Capsule Network Architecture **GAN -** Architecture and Training Methods, Image-Generation, Hands-On Implementation Using Keras. **Transfer Learning** - Motivation, Variations, Use in CNNs.

12 Hours

Unit 5 : Paper Review and Implementation
Selection of two state-of-the-art papers (recent) on deep learning, in depth study of the papers in class and their Implementation – Topics – Meta Learning and Graphical Neural Networks.

10 Hours

NOTE: Unit 5 will be part of End-semester Assessment. Questions will be asked on the chosen papers.

Tools / Languages : Tensorflow 1.15, Keras 2.3.1, Python 3.7.

Text Book:
1. "Advanced Deep Learning with Python" - Ivan Vasilev, Packt Publishing, 2019.

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 for SVM, Transfer Learning.

UE18CS344 : Advanced Computer Networks (4-0-0-4-4)

Advanced Computer Networks course will help students design networks that meet a customer's business and technical goals. This course provides tested processes and tools to help you understand traffic flow, protocol behavior, and internetworking technologies. Students will be equipped to design enterprise networks that meet a customer's requirements for functionality, capacity, performance, availability, scalability, affordability, security, and manageability.

Course Objectives:

- Identify a customer's business and technical requirements for a network design.
- Analyze the existing network and network traffic caused by applications and protocols.
- Develop a topology for a network design.
- Select the right switching, routing and management protocols for network design customer.
- Select technologies and devices for campus and enterprise networks.

Course Outcomes:

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

- Gather a list of customer's most important business and technical goals.
- Design an enhanced network which improves performance of existing network.
- Increase the probability of meeting a customer's goals for scalability, adaptability, and performance.
- Recommend the best switch and router products to the customer.
- Design a campus network or WAN design with all the customer requirements.

Pre-Requisite:UE18CS301 – Computer Networks.

Course Content:

Unit 1: Identifying Your Customer's Needs and Goals

Analyzing Business Goals and Constraints:Using a Top-Down Network Design Methodology, Analyzing Business Goals, Analyzing Business Constraints, Business Goals Checklist. Analyzing Technical Goals and Tradeoffs: Scalability, Availability, Network Performance, Security, Manageability, Usability, Adaptability, Affordability, Making Network Design Tradeoffs, Technical Goals Checklist.

10 Hours

Unit 2: Characterizing the Existing Internetwork and Network Traffic

Characterizing the Network Infrastructure, Checking the Health of the Existing Internetwork, Network Health Checklist. Characterizing Network Traffic: Characterizing Traffic Flow, Characterizing Traffic Load, Characterizing Traffic Behaviour, Characterizing Quality of Service Requirements, Network Traffic Checklist.

12 Hours

Unit 3: Logical Network Design – I

Designing a Network Topology: Hierarchical Network Design, Redundant Network Design Topologies, Modular Network Design, Designing a Campus Network Design Topology, Designing the Enterprise Edge Topology, Secure Network Design Topologies. Designing Models for Addressing and Numbering: Guidelines for Assigning Network Layer Addresses, Using a Hierarchical Model for Assigning Addresses, Designing a Model for Naming.

12 Hours

Unit 4: Logical Network Design – II

Selecting Switching and Routing Protocols: Making Decisions as Part of the Top-Down Network Design Process, Selecting Switching Protocols, Selecting Routing Protocols, IP Routing. Developing Network Management Strategies: Network Management Design, Network Management Architectures, Selecting Network Management Tools and Protocols.

12 Hours

Unit 5: Physical Network Design

Selecting Technologies and Devices for Campus Networks: LAN Cabling Plant Design, LAN Technologies, Selecting Internetworking Devices for a Campus Network Design, Example of a Campus Network Design.

Selecting Technologies and Devices for Enterprise Networks: Remote-Access technologies, Selecting Remote-Access Devices for an Enterprise Network Design, WAN technologies, Example of a WAN design.

10 Hours

Tools / Languages : Claynet, Cisco Packet Tracer.

Text Book:

1. "Top-Down Network Design", Priscilla Oppenheimer, Cisco Press, 3rd Edition, 2011.

Reference Book(s):

1: "Modeling and Tools for Network Simulation", KlausWehrle, Mesut Günes and James Gross, Springer, 2010.
2: "Networking Systems Design and Development", Lee Chao, CRC Press, 2009.
3: "The Practice of System and Network Administration", Thomas A. Limoncelli, Christina J. Hogan and Strata R. Chalup, Addison-Wesley Professional, 2016.

UE18CS345 : 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.

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.
- Provide experience of collaborative work that develops biologically inspired solutions to practical problems.

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.
- Apply nature-inspired algorithms to optimization, design and learning problems.

Pre-Requisite: UE18CS251 – Design and Analysis of Algorithms.

Course Content:

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.

12 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.

12 Hours

Unit 3 : Social Computing: Particle Swarm 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.

12 Hours

Unit 4 : Other Foraging Algorithms

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.

Other Social Algorithms: Glow Worm Algorithm, Bat Algorithm, Fish School Algorithm, Locusts.

10 Hours

Unit 5 : Immuno computing: Artificial Immune Systems

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.

10 Hours

Tools/Languages : Matlab.

Text Book:

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.

UE18CS346 : Social Network Analytics (4-0-0-4-4)

Everything is connected: people, information, events and places, all the more so with the advent of online social media. A practical way of making sense of the tangle of connections is to analyze them as networks. The course will explore how to practically analyze large scale network data and how to reason about it through models for network structure and evolution.

Course Objectives:

- Provide students background on the concept of various types and kinds of Social Networks, their structural properties and related measures.
- Train students to observe and measure unique aspects of network formation and growth of social networks.
- Enable students to understand social phenomena such as diffusion and cascades.
- Expose students to Strategic Networks, the incentive model for connection formation.
- Expose students to Game theory and Games on Networks, concepts related to strategies and optimality.

Course Outcomes:

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

- Model a given scenario/problem as a network, evaluate the type and kind of such a network and measure structural properties of that network.
- Apply algorithms to detect communities and decipher phenomena peculiar to social networks such as small worlds and power laws.
- Model a social process such as spread of information and diseases using diffusion model.
- Model and analyze strategic networks and measure network properties.
- Apply Social Network Analysis concepts to a variety of real world scenarios by modelling them as games**.**

Pre-Requisite: UE18CS202 – Data Structures , UE18MA251 - Linear Algebra

Course Content:

Unit 1 : Background and Fundamentals of network analysis

Introduction to Networks and Examples, Graph-Theory, Directions and Weights, Adjacency Matrices, Affiliation Networks, Cliques and Cores, Cohesion, Ego-centric Networks, homophily, Tie-strengths and structural holes**.** Representing and Measuring Networks: Degree distribution, diameters, path-lengths, Centrality measures, Clustering coefficient, Link Analysis: PageRank.

12 Hours

Unit 2 : Models of Network formation

Erdos and Renyi Random Networks: Thresholds and Phase Transitions, Small World: Milgram's small world experiment, Wats-Strogatz small world model, Growth Models: Barabasi and Albert's model, Distribution of expected degrees, Preferential attachment, Power Laws, Zipf's LawandFat Tails, Scale-free networks.

12 Hours

Unit 3 : Community Detection:

Community Structure, Girvan Newman and Louvain algorithms, finding Overlapped Communities by CPM. **Implications of Network Structure:** Diffusion through Networks: The Bass Model, Diffusion in Random networks, Giant Components, Models to study disease and information spreads, Cascades and Contagions, Percolation and Robustness of Networks, Effects of communities and centralities on diffusion.

12 Hours

Unit 4 : Strategic Networks

Economic Game Theoretic Models of Network Formation, Connections Model, Pair-wise Stability, Efficient and Pareto-efficient networks, Pair-wise Nash Stability, Externalities and Co-author Models, Complements and Substitutes.

10 Hours

Unit 5 : Games on Networks

Introduction to Games, Reasoning about behavior in a Game, Prisoner's Dilemma, Best response and Dominant Strategies, Nash Equilibrium, Multiple equilibriums: Co-ordination Games, Hawk-Dove Game, Mixed Strategies, Pareto Optimality and Social Optimality**.**

10 Hours

Tools / Languages : Gephi, VNetLogo, NetworkX, SocNetV.

Text Book:

1. "Networks, Crowds, and Markets: Reasoning About a Highly Connected World", D Easley and J Kleinberg, Cambridge University Press, 2010.

Reference Book(s):

1: "Social and Economic Networks", Mathew O Jackson, Princeton University Press, 2010.
2: "Networks – An introduction", MEJ Neumann, Oxford University Press 2010.
3: "Analyzing the social web", Jennifer Golbeck, Morgan Kaufmann, 2013.

UE18CS347 : Information Security (4-0-0-4-4)

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

Course Objectives:

- Understand various cyber threats and attacks, learn about Secure Software development process.
- Understand how privilege escalation attacks and Buffer overflows happen Learn to analyze malware and differentiate between categories such as Virus and Worms.
- Understand the concept of Threat Modelling and its application.
- Understand and learn about the most common Web application security vulnerabilities.
- Understand and apply various security testing techniques and tools.

Course Outcomes:

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

- Be able to Identify possible misuse cases in the context of software development.
- Apply the different concepts and techniques learnt to prevent privilege escalation and Buffer overflow attacks. Apply the acquired knowledge to Create a Worm from a Virus.
- Apply Threat Modelling techniques to expose inherent/dormant vulnerabilities in a given software design /architecture and propose alternate solutions.
- Be able to design and develop Secure Web applications.
- Be able to perform Penetration testing on the given software system.

Course Content:

Unit 1 : Introduction

Software Threats and Vulnerabilities, OWASP Top 10, SANS Top 25, CVE, etc. The CIA Triad - Core Security Principles, Vulnerabilities, Threats and Attacks, 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 Software Development Life Cycle, Risk analysis, SDL, SDL practices.

12 Hours

Unit 2 : Privilege Escalation Attacks

Set UID program and environment variable, Shell shock attack, buffer overflow. Non-executable stack, defeat countermeasures, Environment Setup, Tasks involved in the attack, Function prologue and epilogue, Format string Vulnerabilities, Vulnerabilities code launching attack. Malware and abra worm, Stuxnet worm, Morris worm.

12 Hours

Unit 3 : Threat Modelling

Threat Modelling, Trust Boundaries Threat Modelling, Brainstorming, Modelling Methods, stride privacy threats, Taxonomy Of Privacy, privacy tools, processing threats, defensive tactics and technologies. EOP card game.

10 Hours

Unit 4 : Web Application Security Issues

Challenges, browser security, web security. SQL injection. Basic Structure of Web Traffic, Relational Database Elements, SQL Tutorial, Interacting with Database in Web Application, Launching SQL Injection Attacks. Cross Site Request Forgery, CSRF Attacks on HTTP GET / POST Services, Countermeasures, Cross-Site Scripting Attack, XSS Attacks, Countermeasures. HTTP Security: Overview of HTTP Security, MITM Attacks and Solutions, HTTP Security Headers Privacy Issues and HTTP Authentication.

12 Hours

Unit 5 : Security Testing Countermeasures - Tools, Frameworks, and Services:

Static analysis tools, penetration testing, Benefits, Drawbacks, Web hacking Tools, Nmap for network probing, Web proxies, Metasploit, Ethical Hacking, Fuzzing.

10 Hours

Note: Giving hands on experience for relevant topics in the form of Lab or Assignment.
Relevant cyber security Case for undergraduate students are discussed.

Tools / Languages : Seed labs, Scapy ,Burp-suit, N-Map, C- Language.

Text Book:

1. "Computer and Internet Security", Hands on Approach", Wenliang Du , 2nd Edition, 2019.

Reference Book(s):

1: Computer Security – Principles and Practice", William Stallings, 3rd Edition,2014.

UE18CS348 : 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:

- Familiarize the psychology underlying user-interface and usability design guidelines keeping in mind human behavioral and perceptual capabilities and limitations that affect interface design.
- Familiarize 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.
- Apply development methodologies and life-cycle models for building user interfaces and prototyping in user interface design and how to test them.
- Familiarize with the impact of usable interfaces in the acceptance and performance utilization of information systems.
- Highlight the importance of working in teams and the role of each member within an interface development phase.

Course Outcomes:

- Develop and use a conceptual vocabulary for analysing human interaction with software: affordance, conceptual model, feedback.
- Explain how user-cantered design complements other software process models.
- Use lo-fi (low fidelity) prototyping techniques to gather, and report, user responses.
- Define a user-centred design process that explicitly takes account of the fact that the user is not like the developer or their acquaintances.
- Choose appropriate methods to support the development of a specific UI.

Course Content:

Unit 1 : Foundations of HCI

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.

12 hours

Unit 2 : Design and Software processes

Interactive Design basics – 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, rules. Evaluation Techniques – Universal Design.

12 hours

Unit 3 : Models and Theories

Cognitive models –Socio-Organizational issues and stake holder requirements –Communication and collaboration models.

10 hours

Unit 4 : Task Analysis

Task Analysis-Dialog notations and design,Models of the system,Modeling rich interaction.

10 hours

Unit 5 : Outside the Box

Groupware, ubiquotous computing,augmented realities, hyper text, multimedia and World Wide Web.

12 Hours

Text Book :

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

UE17CS401 : Object Oriented Modelling and Design (4-0-0-4-4)

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 Modeling Language (UML) will be used as a tool to demonstrate the analysis and design ideas and an object oriented programming language such as Java would be used to implement the design. The theory is supplemented with a project which provides the hands on experiences of working with teams.

Course Objectives:

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

Course Outcomes:

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

- Use the concepts of classes and objects of object oriented programming in UML to model a complex system.
- Construct static models using use cases and class models followed by analyzing 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 GRASP and SOLID principles in the design of software application.
- Apply different software design patterns in a variety of application development.

Course Content:

Unit 1 : Introduction to Object Oriented Programming and Static Models

Introduction to object oriented concepts using Java. Object-Oriented Analysis and Design: Introduction to UML, Introduction to Modelling, Designing Objects: Static and Dynamic Modelling. Use Case Modeling: Use Cases and the Use-Case Model, Guidelines, Applying UML to Use Case Diagrams, Relating Use Cases. Object Design Technique: CRC Cards. Class Modelling: UML Class Diagrams, Design Class Diagram, Domain Model Refinement.

12 Hours

Unit 2 : Dynamic Models: Sequence Modelling

System Sequence Diagrams (SSDs), Relationship between SSDs and Use Cases, Guidelines. Activity Modelling: UML Activity Diagrams and Modelling, Guidelines. State Modelling: UML State Machine Diagrams and Modelling, Applying UML to State Machine Diagrams. Advanced State Models.

12 Hours

Unit 3 : Architecture Design: Package Design

Package Organization Guidelines. UML Deployment and Component Diagrams.Designing a Persistence Framework with Patterns: UML Data Modelling Profile.Mapping Objects: Database Mapper or Database Broker Pattern.Documenting Architecture: The Software Architecture Design (SAD) and Its Architectural Views. Iterative Development and Agile Project Management.

10 Hours

Unit 4 : Object Oriented Design Principles: GRASP

Designing Objects with Responsibilities, UML versus Design Principles, Object Design, Responsibilities and Responsibility-Driven Design, Connection Between Responsibilities, GRASP and UML Diagrams, Applying GRASP 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. : Introduction to Design Patterns. Implementing Design Patterns: Creational, Structural and Behavioural Patterns.

10 Hours

Unit 5 : Object Oriented Design Patterns and Anti Patterns

Applying Design Patterns: Adapter, GRASP Principles as a Generalization of Other Patterns, Factory, Singleton Strategy, Composite, Facade and Observer/Publish-Subscribe/Delegation Event Model. What are anti-patterns? , Software Development anti-patterns, Software Architecture Anti-patterns, project management anti-patterns.

12 Hours

Tools / Languages : Star UML, Java.

Text Book:

1. "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development", by Craig Larman, 3rd Edition, Pearson 2015.

Reference Book(s):

1: "Java The Complete Reference," 11th Edition by Herbert Schildt, McGraw-Hill 2018.
2: "Object-Oriented Modeling and Design with UML", Michael R Blaha and James R Rumbaugh, 2nd Edition, Pearson 2007.
3: "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, 1st Edition, Pearson 2015.

UE17CS402 : 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 an post college environment. The theory is supplemented with a large project which provides the hands on experiences of working with large teams.

Course Objectives:

- Ensure the relevance and need of an engineering approach to software development.
- Learn Software Engineering concepts.
- Expose students to the tools available as part of the Software Development and Product Development Life Cycle.
- Enable the students to practice the principles of 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 relate to Software Engineering as a methodical approach for development.
- Use Software Development Life Cycles with an understanding of when and where to use.
- Work in different lifecycle phases and produce artifacts expected at each phase, and evaluate using quality metrics.
- Work on a project plan, track and manage projects.
- Understand the connectivity of the development process to operations, and relate to the DevOps activities.

Course Content:

Unit 1 : Introduction to Software Engineering

Understand the Context of Software Engineering, Contrasting System Development, Product Development, Software Products, Project Engineering, Generic Process Framework, Phases in the Development of Software, Product Life Cycle Phases, Roles in Product Development, Product Development Eco-System, Introduction to Software Development Models including Waterfall Model, Incremental Model, Evolutionary Model, Agile Model, CBSE etc. Software Maintenance Lifecycle. **Requirements Engineering and Modelling:** Requirements Engineering Tasks, Requirements Documentation / Specification and Management, Requirements Traceability.

12 Hours

Unit 2 : Software Project and Product Management Overview

Planning a Software Development Project with Overview of Different aspects of Software Engineering Management and Process Maturities. Software Estimation. Exposure to a PM tool like MS Project **Software Architecture:** Software Architecture, Software Life Cycle, Architecture Design, Architectural Views, Architectural Styles, the Unified Modelling Language. **Software Design:** Classical Design Methods, Object Oriented Analysis and Design Methods, Design Patterns, Service Orientation - Service Descriptions and Service Communication, Service Oriented Architecture.

12 Hours

Unit 3 : Implementation

Coding Standards and Guidelines, Code Review / Peer Review. **Change, Build and Release Management:** Elements of a Configuration Management System, Baselines, Repository, SCM Process, Configuration Management Plan, Management of Code Versions, Release Versions, Release management including Patching and Patch Management. Exposure to Code Management Tools like GitHub, BuildBot.

10 Hours

Unit 4 : Software Testing

Test Objectives, Testing & Software Life cycle, Testing Strategies, Verification and Validation, Planning and Documentation, Manual Test Techniques, Coverage Based Test Techniques, Fault Based Test Techniques, Error Based Test Techniques, Comparison of Test Techniques, Test Stages and Estimating Software Reliability. Exposure to test tools Project/Assignment Reviews, presentation by teams.

10 Hours

Unit 5 : Software Quality

Managing Software Quality, A Taxonomy of Quality Attributes, Perspectives on Quality, Metrics, The Quality System, Software Quality Assurance, The Capability Maturity Model, Software Metrics **Other Eng. Topics:** Software Engineering in a Global Environment, Software Engineering and Hacking, Ethics in Software Engineering. **ITSM, ITIL DevOps:**Stitching it all together, Pillars of DevOps – Collaboration, Affinity, Tools and Scaling, Continuous Build, Integration, delivery, Automated Build, testing and deployment. Exploration to a tool like Jenkins.

12 Hours

Tools / Languages : GitHub, MS Project, Jupiter etc.

Text Book(s):

1. “Software Engineering: Principles and Practice", Hans van Vliet, Wiley India, 3rd Edition, 2010.

Reference Book(s):

1: “Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling” by By Jennifer Davis, Ryn Daniels
2: “Software Engineering: A Practitioner’s Approach”, Roger S Pressman, McGraw Hill, 6th Edition 2005
3: “Software Engineering”, International Computer Science Series, Ian Somerville, Pearson Education, 9th Edition, 2009.
4: IEEE SWEBOK and Other Sources from Internet.

UE17CS411 : Enterprise and Resource Planning(4-0-0-4-4)

ERP is an undergraduate engineering course that deals with Enterprise Resource Planning software application solution that is utilized in organizations of various industries across the world. ERP addresses the enterprise needs of an organization by tightly integrating the various functions of an organization using a process view of the organization. In this course important internal sub-systems are studied in addition to the study of the life cycle of ERP implementation in medium to large enterprises.

Course Objectives:
- To learn the strategic importance of Enterprise Resource Planning systems in industry.
- To learn the basics of ERP, costs and benefits and the modules of ERP.
- To learn about Change Management, BPR and BPM.
- To learn key selections criteria, issues & risks involved in ERP implementation.
- To be aware of ERP related technologies and commercial ERP software.

Course Outcomes:
At the end of the course, the student will be able to:
- Identify typical functionality of ERP sub-systems.
- Apply strategies for Change Management, BPR and BPM.
- Apply criteria to select ERP Package and Consulting Partner.
- Systematically develop plans for an ERP Implementation project and.
- Identify critical success factors and associated risks.

Course Content:
Unit 1 : Introduction to Enterprise Resource Planning Systems
Overview, Definition of ERP, Evolution of ERP Systems, ERP System Benefits and Challenges, Enterprise Processes, Extended ERP, Major ERP Players – Product and Consulting Companies, ERP Implementations. **ERP Implementation Life Cycle:** Life Cycle of an ERP Implementation Project, Phases, Methodologies, Types of ERP Projects, Deployment Strategies. **Business Case and ROI for ERP System:** Benefits and Costs of an ERP Implementation, Cost-Benefit Analysis.

12 Hours

Unit 2 : Change Management
Reasons People Resist Change, Change Management Strategies, Organization Design, Change Management Team and Roles, Change Management Activities. Business Process Re-Engineering: Principles and Need, Definition, Phases, Pros and Cons, Keys to Success, Reasons for Failure, BPR Team and Roles, Process Selection and Diagnosis, Process Redesign, BPR and ERP, Benchmarking, Best Practices. Business Process Modelling and Business Modelling: BPM Need, Guidelines, As-Is and To-Be Modelling, Business Process Hierarchy, Standards for BPM, Process Modelling Software, Business Modelling, Integrated Data Modelling.

12 Hours

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

12 Hours

Unit 4 : **Procurement and Inventory Management**
Procurement Process, Types, KPIs. Inventory Management Process, Types, Models, KPIs. **ERP Package Selection:** Selection Team, Selection Criteria, Parameters for Package Selection, Request for Proposal (RFP), Gap Analysis, ERP Market. **ERP Consulting Partner Selection:** Selection Criteria, RFP Process, In-house and Offshore Implementations, Pros and Cons. **Managing an ERP Project:** Scoping, Plan, Charter, Risk Management. Project Teams. Success or Failure of an ERP Implementation.

10 hours

Unit 5:ERP and Enterprise Applications
Supply Chain Management (SCM), Customer Relationship Management (CRM), Product Life Cycle Management (PLM), Data Warehousing, Business Intelligence (DW-BI), ERP on Cloud, ERP for Manufacturing and Service Industries. Articles and Case Studies.

10 hours

Text Book:

1. “Enterprise Resource Planning- Text & Cases”, Rajesh Ray, McGraw Hill Education, New Delhi, 2017.

Reference Book(s):

1: “ERP Demystified”, Alexis Leon, McGraw Hill Education, 3rd Edition, 2014.
2: “Enterprise Resource Planning: A Managerial Perspective”, Veena Bansal, Pearson Education India, 2013.

UE17CS412: Algorithms for Information Retrieval (4-0-0-4-4)

This course deals with the challenge of retrieving information from text data that matches the information need of the user. It starts with the basic building blocks of search such as dictionary and index implementation, compression of such components, ranked retrieval and then methodologies to evaluate and extend the same. It then discusses alternative models of IR and delves into Web IR along with its components and algorithms. Finally it focusses on the various machine learning algorithms and applications that are integral parts of Information Retrieval

Course Objective:

- Understand basic building blocks of search such as Boolean Retrieval, formation of Dictionary and Inverted Index, Phrase Query and Tolerant Retrieval
- Understand basic building blocks of search such as construction of Inverted Indexes, Compression of Dictionary and Scoring Mechanisms for ranked retrieval.
- Understand how a ranked retrieval can be evaluated and how the same can be extended by techniques such as relevance feedback and query expansion. Also explore other Information Retrieval models.
- Learn how Web Information Retrieval is using the techniques such as crawling and link analysis algorithms along with the Economic Model of Web Search
- Learn various machine learning components and applications in the end to end Information Retrieval model.

Course Outcomes:

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

- Get a grip on search building blocks such as Boolean Retrieval, Dictionary, Inverted Index, Phrase Query and Tolerant Retrieval.
- Get a grip on searches performed on a document collection, rank and evaluation of IR.
- Get a firm understanding of evaluation approaches for ranked retrieval, relevance feedback and query expansion mechanisms along with alternative IR methods.
- Get a firm understanding of Web IR – crawler architecture and usage of various algorithms for link analysis instrumental in ranked retrieval.
- Implement machine learning components and applications in Information Retrieval.

Pre-requisite Courses:UE17CS303– Machine Learning

Course Content:

Unit 1: 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.

10 Hours

Unit 2: **Indexing and Vector Space Model**

Algorithms for Indexing and Index Compression, Vector Space Model for Scoring, tf-idf and Variants, Efficient Scoring and Ranking.

12 hours

Unit 3: Evaluation of IR / Other IR Models

Peformance Measurement, Relevance Feedback, Query Expansion, Other IR Models**.**

10 hours

Unit 4: **Web Search**

Web Search Basics, Economic Model of Web Search, Search User Experience, Web Crawling, Link Analysis, The PageRank Algorithm, Building a Complete Search System.

12 hours

Unit 5: Applications of IR

Text Classification and Clustering, Text Summarization, Topic Detection, Question Answering, Snippet Generation, Personalization.

12 hours

Tools / Languages : NLP and ML Libraries / Python 3.x.

Text Book:

1. “Introduction to Information Retrieval”, Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, ISBN: 9781107666399, Cambridge University Press, 2009.

Reference Book(s):

1: “Speech and Natural Language Processing”, Daniel Jurafsky and James H. Martin, 2nd edition paperback, 2013. The more uptodate 3rdedition draft is available at http://web.stanford.edu/~jurafsky/slp3/ .

UE17CS413: Wireless Network Communication (4-0-0-4-4)

Wireless Networks Communication is a dynamic field that has spurred tremendous excitement and technological advances. This course provides a comprehensive understanding of the fundamental principles, characteristics, performance limits of wireless systems, their security issues and the insights associated with their design.

Course Objectives:

- Introduce the emerging trends of wireless network technologies
- Compare and contrast the wireless network technologies depending on the usage models
- Explain the characteristics of wireless channels and analyze its impact during communication
- Discuss various design parameters of communication.
- Identify different attacks on wireless network and explore several mitigation approaches.

Course Outcomes:

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

- Identify and Apply the appropriate wireless technology for real time applications.
- Simulate the channel characteristics such as path loss, shadowing, analyze the wireless networks and understand the Multipath channel models.
- Analyze emerging enhancements such as Adaptive Modulation and Multiple Input Multiple Output System.
- Capture the transmitted packets of wireless networks and analyze them for the wireless communication protocols
- Determine the threats on wireless network and Apply wireless security mechanisms.

Pre-requisite Course: UE17CS301 – Computer Networks.

Unit 1: Overview of Wireless communication

Introduction, Wireless LAN Technology: IEEE 802.11, WPAN Technologies: Bluetooth, Zigbee, NFC, 6LOWPAN, LPWAN- LORA, Weightless, Wireless Local Loop (WLL)-LMDS, MMDS, WiMAX, Long Range Communication- Satellite Communication, Wireless communication analysis using Wireshark.

12 Hour

Unit 2: Overview of Wireless Communication-II

Cellular Network: Cellular System Fundamentals, Channel Reuse, SIR and User Capacity, Interference Reduction Techniques, Mobile Applications and Mobile IP, Tradeoff between Battery, Bandwidth and Distance, Wireless Channel Models: Path Loss and Shadowing Models, Millimeter Wave Propagation, Fading Models: Statistical Fading Models, Narrowband Fading, Wideband Fading Models.

12 Hours

Unit 3: Impact of Fading and ISI on Wireless Performance

Capacity of Wireless Channels, Digital Modulation and its Performance, Adaptive Modulation: Adaptive Transmission System, Adaptive Techniques; Variable-Rate Variable-Power MQAM, Multiple Input/ Multiple Output (MIMO): Narrowband MIMO Model, Parallel Decomposition of the MIMO Channel, MIMO channel capacity, MIMO Diversity Gain: Beamforming, Diversity/Multiplexing Tradeoffs, Space-Time Modulation, Frequency-Selective MIMO Channels, Smart Antennas.

12 hours

Unit 4: Multicarrier Systems

Data Transmission using Multiple Carriers, Multicarrier Modulation with Overlapping Subchannels, OFDM, OFDMA, Single- carrier FDMA, Challenges in Multicarrier Systems, Multiuser Channels: The Uplink and Downlink, Multiple Access, Random Access, Downlink (Broadcast) Channel Capacity, Uplink (Multiple Access) Channel Capacity, Uplink/Downlink Duality.

10 Hours

Unit 5: Security
Attacks on Wireless Network, Attacks on Wireless Clients, WEP-Wired Equivalent Privacy Protocol Security, WiFi Security: WiFi Protected Access, WiFi Protected Access 2, WPA2 Wireless Enterprise Network, RADIUS, Handling Rogue Access Points, Theory of Defense for security wireless Networks.

10 Hours

Tools/ Language: Wireshark, Claynet/packetracer.

Text Book

1. “Wireless Communication”, Andrea Goldsmith, First Edition, Cambridge University Press, 2012

Reference Book(s):

1. "Wireless Communication Networks and Systems", by Cory Beard and William Stallings,1st edition, pearson, 2015.
2. "Wireless Network Security: A Beginner’s Guide" by Tyler Wrightson, McGraw-Hill Education; 1 edition, 2012.

UE17CS414 : Blockchain Technologies (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's important to understand why blockchain is different and how it works in comparison with technologies of the past.

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.
- Learn various vulnerabilities and security mechanisms of blockchain.

Course Outcomes:

At the end of the 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.
- Identify the threats of blockchain and deploy security mechanisms.

Pre-requisite Courses: UE17CS202- Data Structures**.**

Course Content:

Unit 1 : Blockchain Introduction:

Key Blockchain Concepts, Nodes, Cryptocurrency, tokens, Public Ledger, Peer to peer Network, Types of blockchain, Permissioned blockchain model, Permission-less blockchain model, Blockchain Construction.

10 Hours

Unit 2 : Cryptography

Machines that encrypted data in the past, Modern encryption, Private and public keys, Hash functions, From blocks to hashes, Hash Pointer, Markle tree, Ledgers, Transactions and trade, The public witness, Computers that witness, Distributed Consensus, Smart contract design, Bitcoin Blockchain Network.

12 hours

Unit 3: The structure of the Network

Consensus algorithm: Proof of Work, Proof of Stake, Delegated Proof of Stake, Proof of Authority, Proof of Elapsed Time, Proof of Capacity, Proof of Space, Proof of Burn, RAFT, PAXOS, Byzantine Fault Tolerance System, PBFT.

12 hours

Unit 4 : Second generation applications of Blockchain technology

Smart contracts: origins and how they function, Creating and deploying smart contracts, Tokens, Token standards,Second generation tokens Decentralized applications, How are DApps constructed?, Decentralized Autonomous Organizations (DAOs), Hyperledger Fabric: Blockchain-as-a-service (BaaS), Architecture and core components, Hyperledger fabric model, Working on hyper ledger and transaction processing.

12 hours

Unit 5 : Blockchain security

Blockchain vulnerabilities, Smart contract vulnerabilities, Blockchain on CIA security triad, Blockchain based DNS security platform, deploying blockchain based DDOS protection.

10 hours

Tools / Languages : Claynet, Python.

Text Book:

1. “Introduction to Blockchain Technology”, Tiana Laurence, 1st edition, Van Haren Publishing, 2019.

Reference Book(s):

1: “Hands-On Cybersecurity 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" by Narayanan, Bonneau, Felten, Miller and Goldfeder, Princeton University Press (19 July 2016).

UE17CS421 : Information Security (4-0-0-4-4)

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

Course Objectives:

- Understand various cyber threats and attacks, learn about Secure Software development process.
- Understand how privilege escalation attacks and Buffer overflows happen Learn to analyze malware and differentiate between categories such as Virus and Worms.
- Understand the concept of Threat Modelling and its application.
- Understand and learn about the most common Web application security vulnerabilities.
- Understand and apply various security testing techniques and tools.

Course Outcomes:

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

- Be able to Identify possible misuse cases in the context of software development.
- Apply the different concepts and techniques learnt to prevent privilege escalation and Buffer overflow attacks. Apply the acquired knowledge to Create a Worm from a Virus.
- Apply Threat Modelling techniques to expose inherent/dormant vulnerabilities in a given software design /architecture and propose alternate solutions.
- Be able to design and develop Secure Web applications.
- Be able to perform Penetration testing on the given software system.

Pre-requisite : UE17CS331 – Computer Network Security.

Course Content:

Unit 1 : Introduction

Software Threats and Vulnerabilities, OWASP Top 10, SANS Top 25, CVE, etc. The CIA Triad - Core Security Principles, Vulnerabilities, Threats and Attacks, 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 Software Development Life Cycle, Risk analysis, SDL, SDL practices.

12 Hours

Unit 2 : Privilege Escalation Attacks

Set UID program and environment variable, Shell shock attack, buffer overflow. Non-executable stack, defeat countermeasures, Environment Setup, Tasks involved in the attack, Function prologue and epilogue, Format string Vulnerabilities, Vulnerabilities code launching attack. Malware and abra worm, Stuxnet worm, Morris worm.

12 Hours

Unit 3 : Threat Modelling

Threat Modelling, Trust Boundaries Threat Modelling, Brainstorming, Modelling Methods, stride
privacy threats, Taxonomy Of Privacy, privacy tools, processing threats, defensive tactics and technologies. EOP card game.

10 Hours

Unit 4 : Web Application Security Issues

Challenges, browser security, web security. SQL injection. Basic Structure of Web Traffic, Relational Database Elements, SQL Tutorial, Interacting with Database in Web Application, Launching SQL Injection Attacks. Cross Site Request Forgery, CSRF Attacks on HTTP GET / POST Services, Countermeasures, Cross-Site Scripting Attack, XSS Attacks, Countermeasures. HTTP Security: Overview of HTTP Security, MITM Attacks and Solutions, HTTP Security Headers Privacy Issues and HTTP Authentication.

12 Hours

Unit 5 : Security Testing Countermeasures - Tools, Frameworks, and Services:

Static analysis tools, penetration testing, Benefits, Drawbacks, Web hacking Tools, Nmap for network probing, Web proxies, Metasploit, Ethical Hacking, Fuzzing.

10 Hours

Note: Giving hands on experience for relevant topics in the form of Lab or Assignment.
Relevant cyber security Case for undergraduate students are discussed.

Tools / Languages : Seed labs, Scapy ,Burp-suit, N-Map, C- Language.

Text Book:

1. "Computer and Internet Security", Hands on Approach", Wenliang Du , 2nd Edition, 2019.

Reference Book(s):

1: Computer Security – Principles and Practice", William Stallings, 3rd Edition,2014.

UE17CS422 : Social Network Analytics (4-0-0-4-4)

Everything is connected: people, information, events and places, all the more so with the advent of online social media. A practical way of making sense of the tangle of connections is to analyze them as networks. The course will explore how to practically analyze large scale network data and how to reason about it through models for network structure and evolution.

Course Objectives:

- Provide students background on the concept of various types and kinds of Social Networks, their structural properties and related measures.
- Train students to observe and measure unique aspects of network formation and growth of social networks.
- Enable students to understand social phenomena such as diffusion and cascades.
- Expose students to Strategic Networks, the incentive model for connection formation.
- Expose students to Game theory and Games on Networks, concepts related to strategies and optimality.

Course Outcomes:

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

- Model a given scenario/problem as a network, evaluate the type and kind of such a network and measure structural properties of that network.
- Apply algorithms to detect communities and decipher phenomena peculiar to social networks such as small worlds and power laws.
- Model a social process such as spread of information and diseases using diffusion model.
- Model and analyze strategic networks and measure network properties.
- Apply Social Network Analysis concepts to a variety of real world scenarios by modelling them as games.

Pre-requisite Courses:UE17CS202 – Data Structures, UE17MA251 - Linear Algebra.

Course Content:

Unit 1 : Background and Fundamentals of network analysis

Introduction to Networks and Examples, Graph-Theory, Directions and Weights, Adjacency Matrices, Affiliation Networks, Cliques and Cores, Cohesion, Ego-centric Networks, homophily, Tie-strengths and structural holes. Representing and Measuring Networks: Degree distribution, diameters, path-lengths, Centrality measures, Clustering coefficient, Link Analysis: PageRank.

12 Hours

Unit 2 : Models of Network formation

Erdos and Renyi Random Networks: Thresholds and Phase Transitions, Small World: Milgram's small world experiment, Wats-Strogatz small world model, Growth Models: Barabasi and Albert's model, Distribution of expected degrees, Preferential attachment, Power Laws, Zipf's Law and Fat Tails, Scale-free networks.

12 Hours

Unit 3 : Community Detection

Community Structure, Girvan Newman and Louvain algorithms, finding Overlapped Communities by CPM. Implications of Network Structure: Diffusion through Networks: The Bass Model, Diffusion in Random networks, Giant Components, Models to study disease and information spreads, Cascades and Contagions, Percolation and Robustness of Networks, Effects of communities and centralities on diffusion.

12 Hours

Unit 4 : Strategic Networks

Economic Game Theoretic Models of Network Formation, Connections Model, Pair-wise Stability, Efficient and Pareto-efficient networks, Pair-wise Nash Stability, Externalities and Co-author Models, Complements and Substitutes.

10 Hours

Unit 5 : Games on Networks

Introduction to Games, Reasoning about behavior in a Game, Prisoner's Dilemma, Best response and Dominant Strategies, Nash Equilibrium, Multiple equilibriums: Co-ordination Games, Hawk-Dove Game, Mixed Strategies, Pareto Optimality and Social Optimality.

10 Hours

Tools / Languages : Gephi, VNetLogo, NetworkX, SocNetV.

Text Book:

1. "Networks, Crowds, and Markets: Reasoning About a Highly Connected World", D Easley and J Kleinberg, Cambridge University Press, 2010.

Reference Book(s):

1."Social and Economic Networks", Mathew O Jackson, Princeton University Press, 2010.

UE17CS423 : Computer Systems Performance Analysis (4–0–0–4–4)

Performance is a key criterion in the design, procurement, and use of computer systems. As such, the goal of computer systems engineers, scientists, analysts, and users is to get the highest performance for a given cost. To achieve that goal, computer systems professionals need, at least, a basic knowledge of performance evaluation terminology and techniques. Anyone associated with computer systems should be able to state the performance requirements of their systems and should be able to compare different alternatives to find the one that best meets their requirements.

Course Objectives:
- Provide students background on the art of system performance evaluation.
- Train students to observe workload, workload selection.
- Enable students to understand capacity planning and benchmarking.
- Enable students to apply experimental analysis and design techniques.
- Enable students to understand different queuing models and apply them.

Course Outcomes:
At the end of the course, the student will be able to:
- Model a given scenario for analysing the performance evaluation of the system.
- Apply workload and workload selection criteria for different scenarios.
- Apply experimental design technique to analyse / predict performance.
- Model and analyze capacity planning and benchmarking.
- Apply queuing models for different scenarios for predicting the performance.

Pre–requisite Courses: UE17CS253-Microprocessor and Computer Architecture, UE17CS302-Introduction to Operating System.

Course Content:
Unit 1 : Introduction
The Art of Performance Evaluation, Common Mistakes in Performance Evaluation, A Systematic Approach to Performance Evaluation, Selecting an Evaluation Technique, Selecting Performance Metrics, commonly used Performance Metrics, Utility Classification of Performance Metrics, Setting Performance Requirements.

10 Hours

Unit 2 : Workloads, Workload Selection and Characterization
Types of Workloads, Addition Instructions, Instruction Mixes, Kernels, Synthetic Programs, Application Benchmarks, Popular Benchmarks, W orkload Selection – Services Exercised, Level of Detail, Representativeness, Timeliness, Other Considerations in Workload Selection, Workload Characterization Techniques – Terminology, Averaging, Specifying Dispersion, Single Parameter Histograms, Multi Parameter Histograms, Principle Component Analysis, Markov Models, Clustering.

12 Hours

Unit 3 : Experimental Design and Analysis, Simulation
Introduction – Terminology, Common Mistakes in Experiments, Types of Experimental Designs, 2k Factorial Designs, Concepts, Computation of Effects, Sign Table Method for Computing Effects, Allocation of Variance, General 2k Factorial Designs, 2kr Factorial Designs, Computation of Effects, Experimental Errors, Allocation of Variation, General Full Factorial Designs with k Factors – Model, Analysis of a General Design, Informal Methods.

12 Hours

Unit 4 : Queuing Models
Introduction – Queuing Notation; Rules for all Queues; Little’s Law, Types of Stochastic Process. Analysis of Single Queue: Birth–Death Processes; M/M/1 Queue; M/M/m Queue; M/M/m/B Queue with Finite

Buffers; Results for Other M/M/1 Queuing Systems. Queuing Networks: Open and Closed Queuing Networks; Product form Networks, Queuing Network Models for Computer Systems. Operational Laws: Utilization Law; Forced Flow Law; Little’s Law; General Response Time Law; Interactive Response Time law; Bottleneck Analysis; Mean Value Analysis and Related Techniques; Analysis of Open Queuing Networks.

12 Hours

Unit 5 : Simulation, Capacity Planning and Benchmarking, Presentation
Introduction to Simulation, Analysis of Simulation Results, Steps in Capacity Planning and Management, Problems in Capacity Planning, Common Mistakes in Benchmarking, Benchmarking Games. The art of data presentation, Guidelines for preparing Good Charts, Common Mistakes, Decision–makers games, Ratio Game, Strategies for winning a ratio game.

10 Hours

Tools/ Languages : Python, Java.

Text Book:

1. “The Art of Computer Systems Performance Analysis”, Raj Jain, Wiley India, 2007.

Reference Book(s):
1: “Performance Modeling and Design of Computer Systems: Queueing Theory in Action”, Harcho–Balter, etal..., Cambridge University Press, 2013.
2: “Probability and Statistics with Reliability, Queueing and Computer Science Applications”, Kishor S. Trivedi,2nd Edition, Wiley India, 2002.
3: “Modeling and Tools for Network Simulation”, Klaus Wehrle, Mesut Gunes, etal..., Springer, 2010.
4: “Queueing Systems vol I & II”, Wiley India, 1975.
5: “Simulation Modeling and Analysis", Averill M. Law, L. Kleinrock, Tata McGraw–Hill, 2007.

UE17CS424 : 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:

- Familiarize the psychology underlying user-interface and usability design guidelines keeping in mind human behavioral and perceptual capabilities and limitations that affect interface design.
- Familiarize 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.
- Apply development methodologies and life-cycle models for building user interfaces and prototyping in user interface design and how to test them
- Familiarize with the impact of usable interfaces in the acceptance and performance utilization of information systems.
- Highlight the importance of working in teams and the role of each member within an interface development phase.

Course Outcomes:

- Develop and use a conceptual vocabulary for analysing human interaction with software: affordance, conceptual model, feedback.
- Explain how user-cantered design complements other software process models.
- Use lo-fi (low fidelity) prototyping techniques to gather, and report, user responses.
- Define a user-centred design process that explicitly takes account of the fact that the user is not like the developer or their acquaintances.
- Choose appropriate methods to support the development of a specific UI.

Course Content:

Unit 1 : Foundations of HCI

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.

12 hours

Unit 2 : Design and Software processes

Interactive Design basics – 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, rules. Evaluation Techniques – Universal Design.

12 hours

Unit 3 : Models and Theories

Cognitive models –Socio-Organizational issues and stake holder requirements –Communication and collaboration models.

10 hours

Unit 4 : Task Analysis

Task Analysis-Dialog notations and design,Models of the system,Modeling rich interaction

10 hours

Unit 5 : Outside the Box

Groupware, ubiquotous computing,augmented realities, hyper text, multimedia and World Wide Web.

12 Hours

Text Book :

1. “Human Computer Interaction”, 3rd Edition , Dix A., Finlay J., Abowd G. D. and Beale R., 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.
4: “The Design of Everyday Things”, (Revised and Expanded Edition)Don Norman,2013.

UE17CS451 : Software Testing (2-0-0-2-2)

The course introduces various aspects of software testing. It covers software testing life cycle, different types of testing. The course also introduces various tools used for the different types of testing.

Course Objectives:

- Concepts of Software Quality and types of testing.
- Different levels of testing – Unit, Integration, System and Acceptance Testing.
- Non-functional Testing and Regression Testing.
- Software testing tools.
- Advances in testing field like cloud and mobile testing.

Course Outcomes:

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

- Apply the concepts of Quality Engineering.
- Apply proper testing technique at different phases of development.
- Understand cost of quality.
- Gain hands-on exposure to few testing tools.
- Understand advanced testing topics like cloud and mobile testing.

Course Content:

Unit 1 : Introduction to Software Quality and Testing

Introduction to Software Quality and its importance. Quality Philosophy and Concepts, Quality Management, Cost of Quality. Verification and Validation. Importance of Testing in different SDLC models. Modified V Model for testing requirements in a project. SQA processes, tools and techniques for Test Life Cycle. Classification of testing types based on method / Requirement / target / needs.

6 Hours

Unit 2 : Unit Testing & Integration Testing

Unit Testing: Definition, Test planning, methodology, code coverage testing. Integration Testing: Overview, Types of integration Testing - top-down, bottom-up, Functional, Bi-directional, System Integration, Scenario Testing. System Testing: Definition, reason and overview. Functional Testing, Test case generation. Static Testing – Manual, Automated (Tool-based), Structural testing – Code complexity testing, Advantages and disadvantages.

6 Hours

Unit 3 : Black Box Testing

Definition and overview, Test Case Design techniques for Black Box Testing, Specification based test design and Requirements Traceability Matrix, Positive and negative testing, Equivalence Partitioning, Boundary Value Analysis, Decision Tables, Advantages and disadvantages. **White Box Testing**: Definition and Overview. Gray.

5 Hours

Unit 4: Acceptance, Non-functional and Regression Testing

Acceptance Testing Overview, Acceptance Testing Approaches and Types. Non Functional Testing: NFT Overview, Scalability, Reliability and Stress testing. Performance Testing Overview, methodology for performance testing. Regression Testing: Definition, Types of regression testing.

6 Hours

Unit-5: Testing Tools

JUnit, JMeter, Monkey Talk, Appium, Robotium, Selenium, Selendroid, UI Automator and Magneto. Defect Management tools. Discuss the tools and do a comparative study. Hands on in a 2 hour lab session.

5 Hours

Tools / Languages : JUnit, JMeter, Selenium.

Text Book:

1. “Software Testing – Principles and Practices” , Srinivasan Desikan and Gopalaswamy Ramesh, Pearson, 2006.

Reference Book(s):
1: " Foundations of Software Testing ", Aditya Mathur, Pearson, 2008.
2: " Software Testing, A Craftsman’s Approach ", Paul C. Jorgensen, Auerbach, 2008.

UE17CS452 : Research Methodology(2-0-0-2-2)

The primary objectives of the course is to develop understanding of the basic framework of research process and to develop an understanding of various research designs and techniques. Also to identify various sources of information for literature review and data collection and to develop an understanding of the ethical dimensions of conducting applied research and to appreciate the components of scholarly writing and evaluate its quality.

Course Objectives:

- Define research and identify the systematic steps to be followed and identify the overall process of designing a research study from inception to its report.
- Students will understand about the various methods of data collection.
- Students will have a knowledge of how to analyse the data.
- Students will gain knowledge of how to write a paper,technical reports and thesis.
- Impart familiarity with ethical issues in educational research, including those issues that arise inusing quantitative and qualitative research.

Course Outcomes:

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

- Understand some basic concepts of research and its methodologies.
- Identify appropriate research topics and data collection method.
- Select and define appropriate research problem and parameters.
- Able to write a paper, technical reports and thesis.
- Prepare a project proposal (to undertake a project).

Course Contents:

Unit 1 : Research

Meaning, Objectives and Characteristics of Research, Research methods Vs Methodology, Types of Research ,Research Process, Criteria of good research -Problem Statement, Literature Survey, Importance, Sources, Identifying Gap Areas.

5 Hours

Unit 2 : Research Design

Basic Principles; Features of Good Design, Methods; Developing a Research Plan, Determining Experimental and Sample Designs, Sampling. **Methods of data collection:** Collection of Primary Data, Observation Method, Collection of Data through Questionnaires, Collection of Data through Schedules, Difference between Questionnaires and Schedules, Collection of Secondary Data.

5 Hours

Unit 3 : Testing of hypotheses and Data Analysis

Basic concepts - Procedure for hypotheses testing, flow diagram for hypotheses testing, Important parametric test - Data Preparation – Univariate analysis (frequency tables, bar charts, pie charts, percentages), Bivariate analysis – Cross tabulations and Chi-square test including testing hypothesis of association.

Results and Discussion: Discussion, Purpose and Function of Discussion, Summary and Conclusions, Abstract-Key Words; References; Citation Styles.

6 Hours

Unit 4 : Report and Article Writing

Structure and Components; Types of Report; Technical Reports and Thesis; Significance; Preparation; Layout, Structure and Language of Typical Reports;

Scientific Article Writing: Title Preparation; List of Authors and Addresses, Abstracts; Economy of Words; Journals in Computer Science, Impact factor of Journals.

5 Hours

Unit 5 : Research Proposal Fundamentals

Grant Proposal, Proposal Parts; Hazards to Good Scientific Practice; Scientific Misconduct, Ethical standards: Association for computing machinery-IEEE code of Ethics-Ten commandments of computer Ethics- Certification Bodies Ethics-Intellectual Property Rights, Patents, Copyrights, Trademarks.

Use of tools / techniques for Research: Methods to search required information effectively, Reference Management Software like Zotero/Mendeley, Software for paper formatting like LaTeX/MS Office, Software for detection of Plagiarism, Ethical issues related to publishing, Plagiarism and Self-Plagiarism.

7 Hours

Tools / Languages : Reference Management Software like Mendeley, Software for paper formatting like LaTeX/MS Office, Software for detection of Plagiarism.

Text Book:

1. **"**Research Methodology:Methods and Techniques" ,C R Kothari,Gaurav Garg,New Age International (p) Limited Publishers,New Delhi,Fourth Multi Colour Edition,2019.

Reference Book(s):

1: "Research Methods for Engineers", David V Thiel, Cambridge University Press, 2014.
2: "Research Methods for Cybersecurity",Thomas W. Edgar and David O. Manz, Elsevier.
3: Reference from internet.