PROGRAM EDUCATIONAL OBJECTIVES

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

PROGRAM OUTCOMES

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

UE19CS101:

INTRODUCTION TO COMPUTING USING PYTHON (4-0-0-0-4)

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

- Learn basics of computer hardware and programming.
- Learn how to solve a given problem.
- Learn various paradigms of programming.
- Learn Python as a programming language.
- Learn how to combine data structures and functions available in Python to solve problems.

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

- Outline the process involved in executing a computer program.
- Program effectively using Python programming language.
- Think using different paradigms of programming.

Course Content:

1. **Introduction**: Computational 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.
2. **Process of Computational Problem Solving**: Introduction to Python Programming Language, Output Function, Variables, Types, id Operators and Expressions, Control Structures, Lists, Dictionaries, Sets, Tuples and Strings.
3. **Functions**: Definition, Call, Positional and Keyword Parameters, Default Parameters, Variable Number of Arguments, Modules - Import Mechanisms, Functional Programming - map, filter, reduce, max, min and lambda functions, List Comprehension.
4. **Object Oriented Programming:** Classes and Objects, Inheritance, Polymorphism, Error Handling and Exceptions - try, except and raise, Exception Propagation.
5. **File Processing:** Reading and Writing Files.

Pre-requisite Courses: None.

Reference Book(s):

1. “Introduction to Computer Science Using Python: A Computational Problem-Solving Focus”, Charles Dierbach, John Wiley, 2012.

UE19CS102:
INTRODUCTION TO COMPUTING USING PYTHON LABORATORY (0-0-2-0-1)

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

- Learn basics of computer programming.
- Learn how to solve a given problem.
- Learn to use various paradigms of programming.
- Learn Python as a programming language.
- Learn how to implement data structures and functions available in Python to solve problems.

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.
4. Program to demonstrate Control Structures.
5. Program to demonstrate Control Structures.
6. Program to demonstrate Lists and Tuples.
7. Program to demonstrate Sets and Dictionaries.
8. Program to demonstrate String Related Operations.
9. Program to demonstrate the Usage of Functions.
10. Program to demonstrate Functional Programming.
11. Program to demonstrate Functional Programming.
12. Program to demonstrate File Handling in Python.

Pre-requisite Courses: None.

Reference Book(s):

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

UE19CS151:
PROBLEM SOLVING WITH C (4-0-0-0-4)

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

- 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:

1. **Counting**: Introduction to Programming, Salient Features of ‘C’, Program Structure, Variables, Data Types, Operators and Expressions, Control Structures, Input/ Output Functions.
2. **Text Processing and String Manipulation**: Single Character Input and Output, Arrays and Pointers, Strings, String Manipulation.

3. **Prioritized Scheduling**: Functions, Structures and Unions, Dynamic Memory Management, Lists, Priority Queue.
4. **Sorting**: Sorting, Combination of Structures, Arrays and Pointers, Callback, Sorting using Callback.
5. File Handling, Enums, Bit Fields, Storage Class, Qualifiers, Life and Scope, Pre-Processor Directives, Conditional Compilation, Pragmas.

Pre-requisite Courses: None.

Reference Book(s):

1. "How To Solve It By Computer", R G Dromey, Pearson, 2011.
2. "The C Programming Language", Brian Kernighan and Dennis Ritchie, Prentice Hall PTR, 2nd Edition, 1988.

UE19CS152:
PROBLEM SOLVING WITH C LABORATORY (0-0-2-0-1)

Course Objectives:

The objective(s) of this course is to,

- 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'.

Pre-requisite Courses: None.

Reference Book(s):

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

PES University.

UE18CS201
DIGITAL DESIGN AND COMPUTER ORGANIZATION (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Fundamental (combinational and sequential) building blocks of digital logic circuits.
- Design of more complex logic circuits such as adders, multipliers and register files.
- Construction, using above logic circuits, of a microprocessor, and its functioning at the clock cycle level.
- I/O subsystem organization and operation.

Course Outcomes:

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

- Perform analysis of given synchronous digital logic circuit.
- Design and implement small to medium scale digital logic circuits from given specification.
- Understand hardware level microprocessor operation and I/O organization, providing a foundation for the higher layers.

Course Content:

1. **Combinational and Sequential Logic Design:** Introduction, Boolean Equations, Karnaugh Maps, Combinational Building Blocks, Synchronous Logic Design.
2. **Digital Building Blocks:** Introduction, Finite State Machines, Parallelism, Arithmetic Circuits, Wallace tree, Shift Add/Multiplier Design, Floating Point Numbers and operations, Sequential Building Blocks.
3. **Architecture:** Introduction, Assembly Language, Machine Language, Programming, Addressing Modes, Compiling, Assembling, Loading.
4. **Micro-architecture:** Introduction, Performance Analysis, Multi-Cycle Processor, Memory Arrays, Logic Arrays.
5. **I/O Organization:** Accessing I/O devices, Interrupts, DMA, Buses, Interface Circuits

Pre-requisite Courses: None.

Reference Book(s):

1. “Digital Design”, M Morris Mano, Michael D Ciletti, Pearson, 5th Edition.
2. “Computer Organization and Design”, David A Patterson, John L Hennessey, Elsevier, 4th Edition.
3. “Digital Design and Computer Architecture”, David Money Harris, Sarah L Harris, Elsevier, 2nd Edition.
4. “Computer Organization”, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Mc Graw Hill, 5th Edition.

UE18CS202:
DATA STRUCTURES (4-0-0-0-4)

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

- Enable the learner with the concepts of recursion and linear data structures viz., Linked Lists, Stacks and Queues.
- Enable the learner with the concepts of non-linear data structures viz., Graphs, Trees, Heaps, Trie and Hashing.
- Hone the learner such that they obtain the ability to compare different implementations of data structures and recognize the advantages and disadvantages of different implementations.
- Inculcate in the learner, the aspects of choosing the appropriate data structure and algorithm design method for a specified application and with the usage of standard libraries.

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

- Implement fundamental data structures viz., Lists, Stacks, Queues, Linked Lists, Binary Trees from first principles
- Demonstrate the use of appropriate data structures for a given problem.
- Design and implement solutions to basic practical problems using customized data structures.
- Develop quick solutions to practical problems using abstract data types.

Course Content:

1. **Data Structures Overview**: Recursion, Pointers, Programming Practices. **Lists**: Definition, Create, Insert, Delete, Update, Traverse and Position-based Operations, Linked List and Array Implementations, Concatenate, Merge, and Reverse Lists, Doubly-Linked List Implementation and Operations, Circular Lists and Multi-List, Applications of Lists.
2. **Stacks**: Definition, Operations, Implementation using Linked-List and Arrays, Applications of Stacks – Postfix Conversion and Expression Evaluation, Parentheses Balancing. **Queues**: Definition, Operations, Implementation, Applications, Circular Queue, Dequeue.
3. **Graphs:** Representation of Graphs - Adjacency/ Cost Matrix, Adjacency Lists, and Traversal of Graphs. **Trees**: General Tree Representation, Traversals, Applications. **Binary Trees**: Definition, Properties, Implementation, Traversals, Applications.
4. **Binary Search Tree**: Definition, Implementation, Search, Insert, Delete operations. Building and Evaluating Binary Expression Tree, AVL Tree, Threaded BST. **Heap Tree**: Implementation, Insert, Delete, FindMin operations. Priority Queue using Arrays and Heap.
5. **Tries:** Definition, Implementation, Applications. **Hashing**: Hash Table, Hash Functions, Collision Handling by Open Addressing, Chaining.

Pre-requisite Courses: UE18CS151 – Problem Solving with C.

Reference Book(s):

1. "Data Structures and Program Design in C", Robert Kruse, C L Tondo, Bruce Leung and Shashi Mogalla, PHI, 2nd Edition, 2015.
2. "Data Structures Using C and C++", Tanenbaum, Langsam, Augenstein, Pearson,

2nd Edition, 2015.

UE18CS203:
INTRODUCTION TO DATA SCIENCE (4-0-0-0-4)

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

- Provide insights about the basic roles of a Data Scientist. Develop a greater understanding of the importance of Data Visualization techniques.
- Develop problem-solving skills.
- Make inferences about the population parameters using sample data.
- Test a hypothesis about the population parameters 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 fitted model.

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

- Develop various visualizations of the data in hand and communicate results of analysis effectively (visually and verbally).
- Analyze a real-world problem and solve the same with the knowledge gained from various distributions study.
- Analyze an extremely large data set and perform exploratory data analysis to extract meaningful insights.
- 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:

1. **Introduction to Data Science:** Introduction, **Sampling:** Sampling Methods, Sampling Errors. **Getting and Analyzing Data:** Reading Files, Need for Data Cleaning and Its Basics, Scraping the Web. **Statistics:** Introduction, Types of Statistics. **Data Visualization and Interpretation:** Histogram, Bar Charts, Scatter Plots, Good vs. Bad Visualization.
2. **Random Variables:** Random Variables, Expectation, Functions of Random Variables. **Probability Distributions:** Discrete Distributions (Bernoulli, Binomial, Poisson), Continuous Distributions (Normal), Normal Probability Plots, Student's t Distribution. Derivation of Distributions.
3. **Probability Distributions:** Principles of Point Estimation - Mean Squared Error, Maximum Likelihood Estimate, The Central Limit Theorem and Applications. **Confidence Intervals:** Using Simulation to Construct Confidence Intervals, Interval Estimates for Mean of Large and Small Samples, Factors affecting Margin of Error.
4. **Confidence Intervals:** Interval Estimates for Proportion of Large Samples, Confidence Intervals for the Difference between Two Means, Interval Estimates for Paired Data. **Hypothesis and Inference:** Hypothesis Testing for Population Mean and Population Proportion of Large Samples, Relationship between Hypothesis Tests and Confidence Intervals, Large-Sample Tests for the Difference Between Two Means
5. **Hypothesis and Inference:** Errors in Hypothesis Testing, Power of a Test, Factors Affecting Power of a Test, Distribution Free Tests, Chi-Squared Test. **Simple Linear Regression:** Correlation, Inference on Population Correlation,

Building the Regression Model, Predictions using Regression Models, Residual Plots.

Pre-requisite Courses: None.

Reference Book(s):

1. "Statistics for Engineers and Scientists", William Navidi, McGraw Hill Education, India, 3rd Edition, 2013.
2. Raj Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modelling", Wiley, 2008
3. Sharon L. Lohr, "Sampling- Design and Analysis", Cengage , 2nd edition (stats) , 2010
4. "Data Science from Scratch", Joel Grus, O'Reilly, 1st Edition, 2015.

UE18CS204:
WEB TECHNOLOGIES I (3-0-0-0-3)

Course Objectives:

The objective(s) of this course is to,

- Introduce students to the basics of HTML, CSS, PHP and JavaScript.
- Familiarize students with the Document Object Model and enable them to create dynamic web pages that react to user input.
- Teach students about installing and configuring Apache Server.
- Introduce students to the newer features available as part of the HTML5 standard.

Course Outcomes:

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

- Design visually appealing websites using HTML and CSS.
- Design solutions for programming questions using JavaScript.
- Create dynamic WebPages by manipulating the Document Object Model.
- Setup a web server and host a website with backend support.
- Incorporate the latest HTML5 features in the WebPages designed by them with fallback options wherever required.

Course Content:

1. **Introduction, UI Design and UX :** Internet, WWW, Web Servers and Browsers, URLs, Basic Markup, Images, Hyperlinks, Lists, Tables, Forms

2. **HTML5 and JavaScript:** Local Storage, Web Workers, Offline Web Applications, Drag and Drop, Introduction to Client-Side Scripting, JavaScript Basics, Screen Input and Keyboard Output, Functions, Objects, Hoisting, Arrays, JavaScript Objects
3. **DOM and DOM Events:** Accessing and modifying DOM**,** Events and Event Handlers - Load, Mouse, Synthetic Events, Key and Form Related Events, Event Bubbling, Cookies

4. **Apache:** MIME, http, httpd Server, Request Response Formats Basics, Configuration, Debugging, .htaccess
5. **PHP basics and Introduction to AJAX:** File Handling and System Calls, Strings and Regular Expressions, Arrays, Cookies, Sessions, Functions, Classes, Database Access. **AJAX**: Asynchronous GET/POST using XMLHttpRequest
6.

Pre-requisite Courses: None.

Reference Book(s):

1.“JavaScript Absolute Beginner's Guide”, Kirupa Chinnathambi, Que Publishing, 1st Edition, 2017.
2.“Programming the World Wide Web”, Robert W Sebesta, Pearson, 7th Edition, 2013.
3.“HTML5 Up and Running”, Mark Pilgrim, O’Reilly, 1st Edition, 2012.

UE18CS205
DISCRETE MATHEMATICS AND LOGIC (3-0-0-0-3)

Course Objectives:

The objective(s) of this course is to,

- Develop logical thinking and its application to computer science with emphasis on the importance of proving statements correctly.
- Introduce fundamental discrete structures like Sets, Functions and Relations.
- Introduce Combinational objects and counting techniques.
- Draw similarities between Mathematical Induction and Recurrences and use them to design recursive functions.
- Introduce algebraic structures like Groups, Ring and their usage in coding theory.

Course Outcomes:

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

- Comprehend formal logical arguments.
- Specify and manipulate basic mathematical objects such as Sets, Functions and Relations and will also be able to verify simple mathematical properties that these objects possess.
- Apply basic counting techniques to solve combinatorial problems.
- Design a recursive function by developing a Recurrence and prove its correctness using Mathematical Induction.
- Apply the concepts of algebraic structures in coding theory.

Course Content:

1. **Logic:** Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference.
2. **Sets, Functions and Relations:** Sets and Set Operations, Functions, Relations and Their Properties, Representing Relations, Equivalence Relations, Partial Orderings
3. **Counting:** The Sum and the Product Rules, the Pigeonhole Principle, Permutations and Combinations, Binomial Coefficients.

4. **Induction, Recursion and Recurrence Relations:** Mathematical Induction, Strong Induction, Recurrence Relations. **Graphs:** Definition, Complete Graphs, Regular Graphs, Paths, Connectivity, Euler and Hamilton Graphs.
5. **Algebraic Structures:** The Structure of Algebras, Semi Groups, Monoids and Groups, Homeomorphisms, Normal Subgroups, Coding Theory, Hamming Codes.

Pre-requisite Courses: None.

Reference Book(s):

1. "Discrete Mathematics and its Applications", Kenneth H Rosen, Tata McGraw-Hill, 7 th Edition (Indian adaptation by Kamala Krithivasan), 2011.
2. "Discrete and Combinatorial Mathematics: An Applied Introduction", Grimaldi,Ramana, Pearson, 5 th Edition, 2011.

UE18CS206:
DIGITAL DESIGN AND COMPUTER ORGANIZATION LABORATORY (0-0-2-0-1)

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

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

Pre-requisite Courses: None.

Reference Book(s):

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

UE18CS207:
DATA STRUCTURES LABORATORY (0-0-2-0-1)

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

- Enable the learner with the concepts of recursion and linear data structures viz., Linked Lists, Stacks and Queues.
- Enable the learner with the concepts of non-linear data structures viz., Graphs, Trees, Heaps, Trie and Hashing.
- Hone the learner such that they obtain the ability to compare different implementations of data structures and recognize the advantages and disadvantages of the different implementations.
- Inculcate in the learner, the aspects of choosing the appropriate data structure and algorithm design method for a specified application and with usage of standard library.

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

- Implement fundamental data structures viz., Lists, Stacks, Queues, Linked Lists, Binary Trees from first principles.
- Demonstrate the use of appropriate data structures for a given problem.
- Design and implement solutions to basic practical problems using customized data structures.
- Develop quick and foolproof solutions to practical problems using abstract data types.

Course Content:

1. Implementation of a singly linked list with insert and other operations.
2. Implementation of a singly linked list with delete and other operations.
3. Implementation of a doubly linked list with insert and other operations.
4. Implementation of a doubly linked list with delete and other operations.
5. Implementation of a stack using a singly linked list.
6. Parentheses matching using stack data structure.
7. Infix to Postfix conversion.
8. Implement a queue using a singly linked list.
9. Implement a circular queue using an array.
10. Circular deadly game (Josephus Problem).
11. Implementation of a queue using two stacks.
12. Implementation of a BST and tree traversals methods.
13. Counting leaf and non-leaf nodes in a BST.
14. Construction of a max-heap.
15. Implementation of a Priority Queue using a min-heap.

Pre-requisite Courses: None.

Reference Book(s):

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

UE18CS251:
DESIGN AND ANALYSIS OF ALGORITHMS (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Learn to design and analyze algorithms with an emphasis on the resource utilization in terms of time and space.
- Learn various techniques in development of algorithms so that the effect of problem size and architecture design on the efficiency of the algorithm is appreciated.
- Learn to prove the correctness of algorithms.

Course Outcomes:

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

- Identify the design technique used in an algorithm.
- Design an algorithm for a problem in a known design technique.
- Prove the correctness of an algorithm.
- Analyze the resource utilization of an algorithm in terms of time and space.
- Understand the limits of algorithms and the ways to cope with the limitations.

Course Content:

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.
2. **Brute Force:** Sequential Search, Brute Force String Matching, Selection Sort, Bubble Sort, Depth-First Search and Breadth-First Search, Exhaustive Search. **Divide-and-Conquer:** Merge Sort, Quick Sort, Binary Search, Binary Tree Traversals, Multiplication of Large Integers, Strassen's Matrix Multiplication and Master Theorem.
3. **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.
4. **Space and Time Tradeoffs:** Sorting by Counting, Input Enhancement in String Matching - Horspool's and Boyer-Moore Algorithms. **Dynamic Programming:** Computing a Binomial Coefficient, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms.
5. **Greedy Technique:** Prim's Algorithm, Kruskal's Algorithm and union-find algorithm, Dijkstra's Algorithm, Huffman Trees. **Limitations of Algorithm Power:** Lower-Bound Arguments, Decision Trees, P, NP, and NP-Complete Problems. **Coping with the Limitations of Algorithm Power:** Backtracking, Branch-and-Bound.

Pre-requisite Courses: UE18CS151 – Problem Solving with C.

Reference Book(s):

1. "Introduction to the Design and Analysis of Algorithms", Anany Levitin, Pearson Education, 2nd Edition, 2011.
2. "Introduction to Algorithms", Thomas H Cormen, Charles E Leiserson, Ronald L Rivest and Clifford Stein, Prentice-Hall India, 3rd Edition, 2009.
3. "Fundamentals of Computer Algorithms", Horowitz, Sahni, Rajasekaran, Universities Press, 2/e, 2007.
4. "Algorithm Design", Jon Kleinberg, Eva Tardos, Pearson Education, 2006.

UE18CS252:
DATABASE MANAGEMENT SYSTEMS (3-0-0-0-3)

Course Objectives:

The objective(s) of this course is to,

- Introduce fundamental concepts, terminology and application of relational databases.
- Teach design concepts and creation of relational databases.
- Teach basic and advanced SQL commands.
- Provide overview of database programming and procedural languages.
- Teach normal forms and normalization.

Course Outcomes:

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

- Construct an Entity-Relationship (E-R) model from specifications and transform it to a relational model.
- Design databases and apply normalization techniques.
- Construct queries in SQL and Relational Algebra to perform CRUD (Create, Retrieve, Update and Delete) operations on database.
- Understand and apply the concepts of database programming.
- Design and Build a database application using a RDBMS.

Course Content:

1. **Introduction to Database and Conceptual Design using ERD:** Introduction to Databases**,** Conceptual Model**,** Conceptual Design using ERD, Entity, Weak Entity, Relationships, Attributes and Keys, Roles and Constraints,
2. **Relational Model:**Relational Model, Constraints and Database Schemas, ER to Relational Mapping, Relational Algebra, Unary Operations - SELECT and PROJECT, Set Theory Operations, Binary Relational Operations - JOIN, DIVISION, Aggregate Functions and Grouping.
3. **SQL:** SQL Data Definition, Primary Data Types and Advanced Data Types like CLOB, BLOB, Specifying Constraints in SQL, Basic Retrieval Queries, Insert, Delete, Update and Schema Change Statements in SQL, Advanced SQL Queries, Other SQL Constructs :WITH and CASE, Specifying General Constraints as Assertions and Triggers, Views, Additional Features of SQL, Database Programming, PL/SQL.

4. **Database Design:** Informal Design Guidelines for Schemas, Functional Dependencies, Inference Rules, Closure, Equivalence, Minimal Cover, Normal Forms Based on Primary Keys (1st, 2nd and 3rd NF), General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form, Properties of Relational Decompositions, Overview of Higher Normal Forms.
5. **DBMS Architecture and Database Security:** Three-Schema Architecture, Data Abstraction and Data Independence, Database Languages and Interfaces, DBMS Modules, SQL commands for database transactions (BEGIN, END, COMMIT, SAVEPOINT, ROLLBACK) and Database Security (CREATE USER, ROLE, GRANT and REVOKE).

Pre-requisite Courses: None.

Reference Book(s):

1. "Fundamentals of Database Systems", Ramez Elamsri, Shamkant B Navathe, Pearson, 7th Edition, 2017.
2. "Database Management Systems", Johannes Gehrke, Raghu Ramakrishnan, McGraw-Hill, 3rd Edition, 2003.
3. "Database Systems: The Complete Book", Garcia-Molina, J D Ullman, Widom, Prentice-Hall, 2nd Edition, 2008.
4. "Database System Concepts", Silberschatz, H Korth and S Sudarshan, McGraw-Hill, 6th Edition, 2010.

UE18CS253:
MICROPROCESSOR AND COMPUTER ARCHITECTURE (4-0-0-0-4)

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

- Introduce concepts of basic processor architecture and its design.
- Introduce concepts of pipeline architecture and hazards.
- Bring in the study of memory hierarchy, cache memory and its optimizations.
- Introduce multi-core/ many core processor architecture and programming.

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 multi-core architectures.

Course Content:

1. **Introduction to 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.
2. **Introduction to Pipeline:** 3 - Stage Pipelining, 5 - Stage Pipelining, Pipeline Hazards - Data, Structural and Branch Hazards, Branch Prediction Mechanisms, Performance Metrics, Trends in Technology, Power and Energy in Integrated

Circuits.

3. **Memory Hierarchy:** Mapping Techniques - Fully Associative, Direct Mapped and Set Associative, Cache Performance, Basic Cache Optimizations: I to 4.
4. **Cache Optimizations and Parallel Architecture**: Cache Optimization 5 and 6, Introduction to Parallel Architecture, Parallel Programming models, memory architecture
5. **Advances in Architecture:**, Amdahl's Law, Gustafson Law, Instruction Level Parallelism, Multi-Core Architecture.

Pre-requisite Courses: None.

Reference Book(s):

1. "Computer Organization and Design", Patterson and Hennessey, Morgan Kaufmann**,** 5th Edition
2. "ARM System-on-Chip Architecture", Steve Furber, Pearson India, 2nd Edition, 2015
3. "Computer Architecture: A Quantitative Approach", Hennessey and Patterson, Morgan Kaufmann, 5th Edition.

UE18CS254:
THEORY OF COMPUTATION (3-0-0-0-3)

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

- 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.
- To familiarize students with concepts like Recursively Enumerable languages, Recursive Languages, PCP and Undecidable Problems.

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 other 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.
- 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, Post Correspondence Problem, Undecidable Problem for Context – Free Languages.

Course Content:

1. **Introduction:** Languages, Grammars, & Automata. **Finite Automata:** Deterministic Finite Automata, Non-Deterministic Finite Automata, constructing Finite Automata, equivalence of Deterministic and Non-Deterministic Finite Automata, Minimizing Finite Automata.
2. **Regular Languages and Grammars:** Regular Expressions, Construction, Equivalence of Regular Expressions, Regular Languages and Finite Automata, Regular Expressions in Practice, Regular Grammars, their Construction and Equivalence to Finite Automata. **Properties of Regular Languages**: Closure Properties of Regular Languages, Answering Questions About Regular Languages, Pumping Lemma and identifying Non-Regular Languages.
3. **Context–Free Languages and Grammars:** Context–Free Grammars, Constructing Context–Free Grammars, Parsing and Ambiguity, Simple and Linear Grammars. **Pushdown Automata:** Non-Deterministic Pushdown Automata, Constructing Pushdown Automata, Equivalence of Pushdown Automata and Context–Free Grammars.
4. **Pushdown Automata:** Deterministic Pushdown Automata and Deterministic Context–Free Languages, Conversion of CFG to PDA. **Simplification of Context–Free Grammars & Normal Forms:** Methods for Transforming Grammars, Conversion to Chomsky Normal Form, a Membership Algorithm for Context–Free Languages, Greibach Normal Form. **Properties of Context-Free Languages:** Closure Properties and Questions about Context–Free Languages, Pumping Lemma for Context–Free Languages.
5. **Turing Machines:** The Standard Turing Machine, Constructing Turing Machines, Church–Turing Thesis, Universal Turing Machine. **Hierarchy of Formal Languages and Automata:** Recursive and Recursively Enumerable Languages, Diagonalization, Chomsky Hierarchy. **Limits of Algorithmic Computation:** Post Correspondence Problem, the Halting Problem of Turing Machines, Undecidable Problems.

Pre-requisite Courses: None.

Reference Book(s):

1. “An Introduction to Formal Languages and Automata”, Peter Linz, Jones and Bartlett, New Delhi, India, 5th Edition, 2011.
2. “Theory of Computation”, Michael Sipser, Cengage Learning, New Delhi, India, 2008.
3. “Introduction to Automata Theory, Languages, and Computation”, John E Hopcroft, Rajeev Motwani, Jeffrey D Ullman, Pearson Education, New Delhi, India, 3rd Edition, 2009.
4. “Theory of Computation: A Problem–Solving Approach”, Kavi Mahesh, Wiley India, New Delhi, 2012.

UE18CS255:
DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY (0-0-2-0-1)

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

- Understand the algorithm design techniques including Brute Force, Divide and Conquer, Decrease and Conquer, Transform and Conquer, Dynamic Programming and Greedy technique.
- Make Space and Time Tradeoffs while designing algorithms.
- Analyze the algorithms with absolute running time of the algorithm implementations and compare it with the asymptotic complexity classes of the algorithms.

Course Outcomes:
At the end of the course, the student will be able to:
- Design and implement algorithms of Brute Force Technique.
- Design and implement algorithms with Divide and Conquer technique.
- Design and implement algorithms with Decrease and Conquer and Transform and Conquer techniques.
- Design and implement algorithms with Space and Time Tradeoffs.
- Design and implement optimization algorithms using Dynamic Programming and Greedy technique.

Course Content:
1. Implementation of Brute Force algorithms.
   Sequential Search Algorithm
   Naive String Matching Algorithm
   Selection Sort Algorithm
   Bubble Sort Algorithm
   Exhaustive Search Algorithm for solving the Travelling Salesman Problem
2. Implementation of Divide-and-Conquer algorithms.
   Merge Sort Algorithm
   Binary Search Algorithm
   Quick Sort Algorithm
   Insertion Sort Algorithm
3. Implementation of Transform-and-Conquer algorithms.
   AVL Trees
   Heap Sort Algorithm
   Depth - First Search Algorithm
   Breadth - First Search Algorithm
   Topological Sort.
4. Implementation of Space and Time Trade-off algorithms.
   Distribution Counting Sort Algorithm
   Horspool's Algorithm for String Matching
5. Implementation of Dynamic Programming algorithms.
   Warshall's algorithm
   Floyd's Algorithm
6. Implementation of Greedy Technique algorithms.
   Prim's algorithm
   Dijkstra's algorithm

Pre-requisite Courses: None.

Reference Book(s):
1. Laboratory Manual prepared by Department of Computer Science and Engineering, PES University.

UE18CS256:

MICROPROCESSOR AND COMPUTER ARCHITECTURE LABORATORY (0-0-2-0-1)

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

- 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 Arduino board.
- Learner 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:
At the end of the course, the student will be able to:

- 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/ x86 Processor. Sample programs using Simulator.
2. Programs on ARM/ x86 using Simulator.
3. Programs on ARM/ x86 using Simulator.
4. Case Study - 3 Stage Pipeline using Simulator.
5. Case Study - 5 Stage Pipeline using Simulator.
6. Case Study - Data Hazards using Simulator - RAW, WAR, WAW.
7. Case Study - Memory Performance Analysis using Simulator.
8. Confirmation of Mini Project Titles and Literature Survey.
9. Mini Project.
10. Mini Project.
11. Mini Project Evaluation.
12. Simple Scalar Simulator, Memory Performance, Simcache.

Pre-requisite Courses: None.

Reference Book(s):

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

UE17CS301:

COMPUTER NETWORKS (4-0-0-0-4)

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

- Impart the core principles of Information Communication Technology, which is an inevitable part of the modern Internet, starting from the layered architecture.
- Teach the principles of OSI Layer and introduce sample popular Application Layer protocols - HTTP, DNS.
- Provide an insight into the most widely used Transport Layer protocols - TCP and UDP. Convey dimensions of Network layer, through the Internet Protocol – IPv4 and IPv6 that glues billions of hosts across the globe and associated routing protocols.
- Give an essence of the Data Link Layer, building blocks of Local Area Network and consolidate how all layers are involved to provide a browsing application.
- Provide the experience of use of network tools to imbibe the diagnostic and debugging skills to deal with networking issues.

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

- Sketch the big picture of complex Internet, in terms of building blocks, organized layered architecture.
- Analyze HTTP, DNS and other protocols and use socket programming.
- Illustrate how reliable communication is achieved on a public internet using TCP and select the right transport protocol for a given application.
- Design subnets, configure routers in simulated environment and analyze IP using a protocol analyzer.
- Demonstrate the role of multiple protocols used in all the layers while running a popular application like browsing.

Course Content:

1. **Introduction to Computer Networks and the Internet:** Building Blocks of Communication Networks and Internet, Elements of Network Edge, Access Networks and Physical Media, Network Core, Concepts of Switching, Layered Architecture - Introduction, List of Responsibility and Functions of Each Layer, Introduction to Network Tools such as Wireshark, nc, ssh.
2. **Application Layer:** Network Application Principles, The Web and HTTP - Overview, HTTP Message Format, Web Caching, Cookies and Authentication, DNS Services, DNS Hierarchy, DNS Records, Socket Programming with TCP and UDP.
3. **Transport Layer:** Introduction to Transport Layer Services, UDP Protocol, Principles of Reliable Data Transfer - Stop – N – Wait protocol, Sliding Window Concepts – Go Back N Protocol, TCP Features, Header, Connection Management, Flow Control, Error Control and Congestion Control, TCP Streaming Vs. UDP Message Oriented Delivery.
4. **Network Layer and Internet Protocol:** IPV4 and IPv6 Datagram Format, Fragmentation, Addressing, Subnet Principles, Forwarding Mechanisms, DHCP, NAT, ICMP, ARP, IP Static Routing, Hierarchical Addressing and Route Aggregation, Longest Prefix Match, Introduction to IPTABLES, Introduction to IPV6.

5. **Link Layer and LAN:** Introduction to Link Layer, Introduction to Error Detection and correction, Building Blocks of Local Area Network, LAN Switch - Working Principles, Introduction to MAC Protocols, CSMA/CD, Retrospective: A Web Page Request Case Study, Introduction to Wireless LAN and WPA/ WPA2 and Access Point.

Pre-requisite Courses: None.

Reference Book(s):

1. "Data Communication and Networking", Behrouz A Forouzan, McGraw Hill, 5th Edition.
2. "Computer Networking - A Top - Down Approach", James F Kurose, Keith W, Pearson, 6th Edition, 2012.
3. "Computer Networks - A Top - Down Approach", Behrouz A Forouzan, Firouz Mosharraf, Pearson, Special Indian Edition, 2012.

UE17CS302:
INTRODUCTION TO OPERATING SYSTEMS (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Provide an understanding on the various components of an Operating System.
- The course focuses on fundamental problems and optimal solutions for resource management in operating systems 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 interface for interacting with a contemporary Operating system such as Linux.

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 operating systems such as process management, concurrent processes and threads, memory management and virtual memory.
- Use tools and interface of the operating system.
- Explore design tradeoffs in designing various components of an Operating System.

Course Content:

1. **Introduction and CPU:** What Operating Systems Do? **Introduction to Virtualization of Resources:** CPU/ Memory, Concurrency, Persistence. The Process Abstraction, Process States, Description, Control, API (fork ()/ exec ()). **Scheduling:** Workload Assumptions, Metrics, Types of Scheduling: FIFO, SJF, Response Time, Round Robin, Multi - Level Feedback Queue. **Case Study:** Linux/ Windows/ UNIX Scheduling Algorithms.
2. **Concurrency:** Introduction and Threads, Types of Threads, Multi - Core/ Multi - Threading, Shared Data, **Thread API:** Thread Creation, Completion, Locks, Condition Variables, Compilation. **Mutual Exclusion and Synchronization:** Software Approaches, Principles of Concurrency, Hardware Support, Semaphores,

Message Passing, Readers Writers Problem, pthread Locks. **Deadlocks and Starvation:** Principles of Deadlock, Tools for Detection.

3. **Memory:** Requirements, Partitioning, Paging, Segmentation, Memory API – malloc/ free, Errors. **Virtual Memory**: Hardware and Control Structures, OS Support, Address Translation, Dynamic Relocation, Segmentation, Paging, TLBs, Context Switches, Replacement Policy - LRU, Design Alternatives – Inverted Page Tables, Bigger Pages, Swapping. **Case Study:** Linux/ UNIX Memory Management.
4. **Persistence: I/O Devices** – System Architecture, Canonical Devices/ Protocol – Organization of I/O, CPU Overheads and Interrupts, DMA, OS Design Issues - Device Interaction, Device Driver, Buffering. **Disk Drives:** Performance Parameters **-** Geometry, I/O Time Computation, Disk Scheduling Policies, Data Integrity and Protection – Checksum.
5. **File Systems:** File Organization and Access, Directories, Sharing, Security – Access Controls, Record Allocation, And Secondary Storage Management. **Case Study:** UNIX/ Windows/ Linux File System. **FS Interface:** Creating/ Reading/ Writing, Random Access, fsync (), Renaming, Hard Links and Symbolic Links, Mounting File Systems. **Security:** Intruders and Malicious Software, Buffer Overflow, OS Hardening, Case Study: UNIX/ Windows.

Pre-requisite Courses: UE17CS202 – Data Structures, UE17CS253- Microprocessor and Computer Architecture.

Reference Book(s):

1. "Operating Systems - Internals and Design Principles", William Stallings, Pearson, 9th Edition, 2018.
2. "Operating Systems: Three Easy Pieces", Remzi Arpaci-Dusseau and Andrea Arpaci Dusseau, 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 System Concepts", Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, John Wiley and Sons, 9th Edition, 2013
5. "Operating Systems", Harvey Deitel, Paul Deitel, David Choffnes, Prentice Hall, 3rd Edition
6. "Modern Operating Systems", Andrew S Tannenbaum, Pearson, 3rd Edition

UE17CS303:
MACHINE LEARNING (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Introduce Basics of Matrices, Random Variates and Distributions relevant for the study of Machine Learning Techniques.
- Formulate a well - defined Machine Learning problem with clear metrics.
- Familiarize with techniques for Dimensionality Reduction and Computational Efficiency.
- Understand the notions of Hypotheses Space, Hypotheses Structure and Search.
- Become conversant with types of Machine Learning Algorithms, their applicability and Inductive Bias.

Course Outcomes:

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

- Distinguish categories of Data Attributes, Dimensions, Sample Sizes.
- Acquire a thorough understanding of Supervised, Unsupervised Learning.
- Understand Logistic and Linear Regression and Function Estimation.
- To cluster and classify data.
- Extract Rules and Associations and provide impactful recommendations from data.
- Decide on the data that matters for the learning problem at hand.

Course Content:

1. **Introduction, Classification with Decision trees :** Introduction to ML, 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.
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.
3. **Boosting and Stochastic Models:** Improving performance with Ada-boost, combining weak learners. Bayesian Learning – Bayes theorem, Concept learning, Maximum likelihood, Bayes optimal classifier, Gibbs algorithm, Naïve Bayes classifier, Expectation maximization and Gaussian Mixture Models, Hidden Markov models – discrete Markov processes, 3 basic problems, Finding State sequence, Learning model parameters.
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.
5. **Genetic Algorithms and Optimization Techniques:** Overview: importance of feature engineering in clustering, Genetic Algorithms – Representing hypothesis, Genetic operators and Fitness function and selection, Simple applications of the Genetic Algorithm, application of GA in Decision tree, Genetic Algorithm based clustering, Single Objective and Bi-objective optimization problems using GA, using GA to emulate Gradient descent/ascent. Introduction to PSO and application in Single Objective optimization problems, PSO and bi-objective optimization, using PSO to emulate Gradient descent/ascent.

Pre-requisite Courses: UE17MA251 – Linear Algebra, UE17CS251 – Design and Analysis of Algorithms.

Reference Book(s):

1. “Machine Learning”, Tom Mitchell, McGraw Hill Education (India), 2013.

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

3. “Introduction to Machine Learning”, Ethem Alpaydin, PHI Learning, 2nd Edition, 2019.

4. Appropriate Handouts for relevant topics

UE16CS304:
COMPUTER NETWORKS LABORATORY (0-0-2-0-1)

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

- Make familiar with some of the popular software tools being used in the networking industry.
- Facilitate the student to see the packet and sense the protocol in both virtual and real time environment; as a result understand the computer networks domain.
- Impart the principles of architecting a typical network through design, engineering and configuration.
- Develop the skills of analysing a protocol which would help in troubleshooting a network.
- Create an environment to apply programming knowledge and skills to computer network domain.

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

- Effectively use the industry standard network simulation tool - CISCO PACKET TRACER and analysis tool – Wireshark and the emulator ClayNet.
- Interpret a protocol in terms of syntax, semantics and sequence of actions.
- Design, engineer and test the network in a virtual environment.
- Analyse standard protocols using industry standard protocol analyser.
- Apply algorithmic approach and coding to implement the principles/ protocols of networking.

Course Content:

1. Study the basic networking tools- Wireshark, and basic commands like TCPDUMP, PING, TRACEROUTE, NETCAT.
2. Study the principle of a switched network using the simulator - CISCO PACKET TRACER
3. Study and Visualize the OSI Layered Architecture using an Emulator ClayNet.
4. Understand Persistent and Non Persistent HTTP Connections and Corresponding Performance Impact.
5. Understanding working of HTTP headers: Conditional GET, Cookies and Authentication.
6. Design and Understand and analyse basic topologies in ClayNet.
7. Write a program to create a simple web server - client system using socket programming.
8. IPV4 Addressing
9. IPV6 Addressing
10. ICMP Redirect and Study Unknown Unicast and broadcast flooding and broadcast storm in L2 network.
11. Understanding TTL expiry.
12. Implementation of ARP and MAC Protocols using Claynet.

Pre-requisite Courses: None.

Reference Book(s):

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

UE16CS305:
INTRODUCTION TO OPERATING SYSTEMS LABORATORY (0-0-2-0-1)

Course Objectives:

The objective(s) of this course is to,

- Introduce various interfaces of the Operating system, including system calls.
- Understand the structure of a process, how it is created and scheduled and synchronization between processes/threads
- Understand concepts of memory management such as paging
- Understand concepts of persistent storage of process and interfaces
- 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. Getting used to basic commands on Linux – process creation, monitoring, file system tree, process states, file system commands
2. Build simple client server program to transfer file from client to server
3. Build a simple shell to transfer a file from the server and pipe it through a word count program
4. Build a multithreaded server to transfer files from server to client. Each thread will read one file and send it back.
5. Compile and boot, Boot xv6
6. To add new systems call (for priority of process) to xv6. Add command line to see it.
7. Process management- process scheduling. Modify scheduler to take into account the new process priority and schedule accordingly
8. Implement corner cases in the process management. Should not cause starvation.
9. Memory Management- Physical memory
10. Memory Management- Virtual memory
11. File System on xv6 - structure (first week to understand structure)
12. File system on xv6 – large size (may take 2 weeks)

Pre-requisite Courses: None.

Reference Book(s):

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

UE17CS311:
ADVANCED ALGORITHMS (4-0-0-0-4)

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

- Understand basics of Recurrences and Amortized Complexity Analysis of Data Structures.
- Understand a few String Matching/ Prediction Algorithms and their applications.
- Understand the design strategy of Dynamic Programming.
- Understand some Polynomial and Number Theoretic Algorithms.
- Learn about Randomized Algorithms.

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

- Perform Amortized Analysis on complex data structures.
- Decide usage of Randomized Algorithms for practical intractable problems.
- Compare and evaluate String Matching Algorithms.
- Apply Number Theoretic concepts in applications like Cryptography.
- Solve complex problems using Dynamic Programming.
- Implement an efficient FFT Algorithm.

Course Content:

1. **Basics of Complexity:** Notations, Randomized Algorithms and Amortized Complexity Analysis.
2. **String Algorithms:** String Matching – Boyer–Moore, Rabin–Karp, Finite Automata, and Knuth–Morris–Pratt Algorithms, Suffix Trees - Applications of Suffix Trees, Regular Expression Searches Using Suffix Trees.
3. **Number Theoretic Algorithms**: Modular Arithmetic, RSA Cryptography, Primality Testing and Factorization.
4. **Dynamic Programming**: Elements of Dynamic Programming, Problems - Coin-Row, Rod-Cutting, Matrix-Chain Multiplication, Longest Common Subsequence.
5. **Polynomials and FFT:** Representation of Polynomials, DFT and FFT, Efficient Implementation of FFT.

Pre-requisite Courses: UE17CS251 – Design and Analysis of Algorithms.

Reference Book(s):

1. “Introduction to Algorithms”, T H Cormen, C E Leiserson, R L Rivest and C Stein, PHI, 3rd Edition 2009.
2. “The Algorithm Manual”, Steven Skiena, Springer, ISBN: 9788184898651.

UE17CS312:
DATABASE TECHNOLOGIES (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Acquire knowledge on Parallel and Distributed Databases.
- Learn topics of Data Warehousing for solving analytical data processing problems.
- Learn basics of NoSQL Databases and Big Data systems.
- Learn specialized Application Databases.
- Choose the appropriate database and storage technique.

Course Outcomes:

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

- Analyze issues related to implementing relational database for large datasets.
- Apply Data Warehousing techniques for solving analytical processing requirements.
- Apply Parallel and Distributed Database approach to problems of large databases.
- Select the NoSQL (non-relational database) approach to the “Big-Data” problem.
- Apply specialized databases for advanced applications.

Course Content:

1. **Review of Relational Data Model**: Reporting and Analytical databases, Data Warehousing, OLAP, SQL Analytical Functions, Column Oriented Storage, Introduction to Data Mining.
2. **Parallel and Distributed Databases:** Concepts, Parallel and Distributed databases and issues.
3. **Introduction to NoSQL:** Emergence of NoSQL databases, Characteristics of NoSQL, Categories of NoSQL systems, CAP Theorem. **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).
4. **Introduction to Big Data:** What is Big Data, Hadoop, HDFS, and Apache Spark.
5. **Specialty Databases:** In-Memory Databases for RDBMS (VoltDB) and Key - Value Store (Redis), Time-Series DBMS, Search Engines, Spatial, Temporal, Deductive.

Pre-requisite Courses: UE17CS252 – Database Management Systems.

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. “Data Mining: Concepts and Techniques”, Jiawei Han et. al., Morgan Kaufmann, 3rd Edition, 2011.
4. “NoSQL Distilled”, Pramod J Sadalage and Martin Fowler, Addison Wesley, 2012.
5. “Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement”, Eric Redmond and Jim R. Wilson, O'Reilly, 2012.

UE17CS313:
BIG DATA (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Provide an introduction to Big Data.
- Introduce storage technologies for Big Data.
- Introduce computational issues and infrastructure for Big Data.
- Introduce algorithms for processing Big Data.
- Application of Big Data techniques to various real life problems.

Course Outcomes:

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

- Explore various characteristics of Big Data Problems.
- Understand principles and design alternative 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.
- Apply Big Data techniques are in real life problems through a group based project.

Course Content:

1. **Introduction:** Big Data definition, Challenges and opportunities with Big Data, Data intensive scientific discovery and the role of Big Data, History, MapReduce – Storage (HDFS), Computation model, Case Study: Google. Introduction to sample Big Data Algorithms – matrix multiplication and pagerank.
2. **Big data infrastructures (Compute/Storage):** MapReduce architecture and internals, Overview of Hadoop Ecosystem, Relational operators on MapReduce, case study: HIVE, Other storage - HBase
3. **In memory computation:** Issues with Hadoop, Scala/Python?, Spark introduction and architecture, Spark – DataFrames
4. **Real time analysis:** Streaming analysis use cases, Real time v/s near real time, Streaming Spark, Kafka, Streaming Algorithms - introduction, Case Study.
5. **Advanced Analytics on Big Data :** Introduction of Complexity modelling of Big Data algorithms, Clustering algorithms, Case study : MLLIB, YARN/Mesos, Project

Pre-requisite Courses: UE17CS252 – Database Management Systems.

Reference Book(s):

1. “Hadoop: The Definitive Guide”, Tom White, O'Reilly, 4th edition, 2009.
2. “Big Data Analytics Beyond Hadoop: Real-Time Applications with Storm, Spark, and More Hadoop Alternatives”, Vijay Srinivasa Agneeswaran, Pearson Education, 2014.
3. “Mining of Massive Datasets”, Anand Rajaraman, Jure Leskovec, Jeffrey D. Ullman, Cambridge Press, 2014.
4. “Spark: Cluster Computing with Working Sets”, Zaharia M, Chowdhury M, Franklin MJ, Shekhar S, Stoica I, HotCloud, ACM New York, 2010.
5. “Resilient Distributed Datasets: A Fault-tolerant Abstraction for In-memory Cluster Computing”, Zaharia, Matei, et al., Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012.
6. “Learning Spark”, Matei Zaharia, Patrick Wendell, Andy Konwinski, Holden Karau., O'Reilly Media, 2015.

UE17CS314:
MULTIMEDIA COMPUTING (4-0-0-0-4)

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

- Teach the basic concepts of Multimedia and Hypermedia, World Wide Web and Overview of Multimedia Software Tools and exploring multimedia applications.
- Train students to understand graphics and image data representation, color in image and video and types of video signals and basics of digital audio.
- Teach various text, image and video compression standards.
- Identify the current and future issues related to multimedia technology.
- Identify both theoretical and practical aspects in designing multimedia systems surrounding the emergence of multimedia technologies using contemporary hardware and software technologies.

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

- Apply different compression techniques depending on the multimedia object streams, interpret the various standards for multimedia communications and their features.
- Demonstrate multimedia and its applications to potential clients.
- Identify and describe the function of the general skill sets in the multimedia industry.
- Identify the basic hardware and software requirements for multimedia development and playback.
- Design and develop applications and exercise proper design choices and meet Quality of Service requirements.

Course Content:

1. **Introduction to Multimedia:** What is Multimedia? Multimedia and Hypermedia, World Wide Web, Overview of Multimedia Software Tools. **Graphics and Image Data Representation:** Graphic/ Image Data Types, Popular File Formats. **Color in Image and Video:** Color Science, Color Models in Images, Color Models in Video.
2. **Fundamental Concepts in Video:** Types of Video Signals, Analog Video, Digital Video. **Basics of Digital Audio:** Digitization of Sound, MIDI: Musical Instruments Digital Interface, Quantization and Transmission of Audio.
3. **Lossless Compression Algorithms:** Introduction, Basics of Information Theory, Run Length Coding, Variable Length Coding, Dictionary Based Coding, Arithmetic Coding, Lossless Image Compression**. Lossy Compression Algorithms:** Introduction, Distortion Measures, The Rate-Distortion Theory, Quantization, Transform Coding, Wavelet Based Coding, Wavelet Packets, Embedded Zero Tree of Wavelet Coefficients
4. **Image Compression Standards:** The JPEG Standard**,** the JPEG2000 Standard**,** the JPEG-LS Standard**,** Bi-Level Image Compression Standards**. Basic Video Compression Techniques:** Introduction to Video Compression**,** Video Compression Based on Motion Compensation**,** Search for Motion Vectors**,** H.261**,** H.263.
5. **MPEG Video Coding I – MPEG-1 and 2:** Overview**,** MPEG-1**,** MPEG-2**. MPEG Video Coding II – MPEG-4, 7 and Beyond:** Overview of MPEG-4**,** Object-Based Visual Coding in MPEG-4**,** Synthetic Object Coding in MPEG-4**,** MPEG-4 Object Types, Profiles and Levels**,** MPEG-4 Part10/H.264**,** MPEG-7**,** MPEG-21.

Pre-requisite Courses: None.

Reference Book(s):

1. "Fundamentals of Multimedia", Ze-Nian Li and Mark S. Drew Pearson Education Inc., 2004.

UE17CS315:
PRINCIPLES OF PROGRAMMING LANGUAGES (3-2-0-0-4)

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

- 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 the 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.
- Identify language features required for supporting various paradigms.

Course Content:

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.
2. **Type Checking and Scopes** (continued…): Type Equivalence, Scope, Scope and Lifetime, 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
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.

4. **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.
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.

Pre-requisite Courses: None.

Reference Book(s):

1. "Concepts of Programming Languages", Robert W Sebesta, Pearson Education, 10th Edition, 2012.
2. "Programming Language Pragmatics", Michael L Scott, Elsevier, 3rd Edition, 2009.
3. "Programming Languages Design and Implementation", Pratt and Zelkowitz, Prentice Hall/ Pearson Education, 4th Edition, 2001.

UE17CS321:
COMPUTER GRAPHICS AND VISUALIZATION (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Impart the basics of computer graphics, different graphics systems and applications of computer graphics.
- Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis.
- Introduce the use of geometric transformations on graphics objects and their application in composite form.
- 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 the 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
- Apply Graphics in greater depth to more complex courses like Image Processing, Virtual, Augmented Reality, etc.,

Course Content:

1. **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, Display Considerations.
2. **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, Performance Characteristics Graphics Programming: The Sierpinski Gasket, Programming Two Dimensional Applications. **The OpenGL:** The OpenGL API, Primitives and Attributes, Color, Viewing, Control Functions, The Gasket Program, Polygons and Recursion, The Three-Dimensional Gasket, Plotting Implicit Functions.
3. **Geometric Objects and Transformations-I:** Scalars, Points, and Vectors, Three-Dimensional Primitives, Coordinate Systems and Frames, Modelling a Colored Cube, Affine Transformations, Rotation, Translation and Scaling.
4. **Geometric Objects and Transformations-II:** Geometric Objects and Transformations, Transformation in Homogeneous Coordinates, Concatenation of Transformations, OpenGL Transformation Matrices, Interfaces to Three Dimensional Applications, Quaternions.
5. **Viewing**: Classical and Computer Viewing, Viewing with a Computer, Positioning of the Camera, Simple Projections, Projections in OpenGL. Introduction to animation and rendering, lights and shades

Pre-requisite Courses: UE17CS251 – Design and Analysis of Algorithms.

Reference Book(s):

1. "Interactive Computer Graphics: A Top-Down Approach with OpenGL", Edward Angel, Pearson Education, 5th Edition, 2008.
2. "Computer Graphics Using OpenGL", Francis S Hill Jr. , Stephen M Kelley, PHI, 3rd Edition, 2009.
3. "Computer Graphics", James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes, Pearson Education, 1997.
4. "Computer Graphics - OpenGL Version", Donald Hearn and Pauline Baker, Pearson Education, 3rd Edition, 2004.

UE17CS322:
DATA ANALYTICS (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Provide an insight into data pre-processing, summarization and visualization techniques.
- Teach role of data analytics in business decision making.
- Teach model building and validation using various techniques.
- Develop hands-on experience in a relevant project based on real-life data.
- Teach the skill of effectively communicating the results of data analytics.

Course Outcomes:

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

- Perform exploratory data analysis on a given set of data including visualization techniques.
- Analyze data to infer underlying patterns and formulate recommendations.
- Build time series models and use them for prediction.
- Perform text analysis involving classification and clustering.
- Interpreting Business values from predictive models.

Course Content:

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.
2. **Regression Analysis**: Multiple Regression, Logistic Regression, Lasso regression, Case Studies.
3. **Recommendation Systems**: Collaborative filtering, Knowledge based filtering, Decision trees and Agglomerative clustering with the mention of KNN, SVM, and DB scan(including a brief introduction to text classification/clustering,)Mining of Association Rules and evaluation of recommendation system, Case Studies
4. **Time Series Analysis:** Simple and exponential Smoothing, Trend Analysis, 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.
5. **Advanced Models:** LSA as an application of SVD, Sparse PCA, Discrete Markov Chain, Confounding variables.

Pre-requisite Courses: UE17CS203 – Introduction to Data Science, UE17MA251-Linear Algebra

Text Books:

1. "Data Mining: Concepts and Techniques", Jiawei Han, Micheline Kamber and Jian Pei, The Morgan Kaufmann Series in Data Management Systems, 3rd Edition, 2011.
2. “The Element of Statistical Learning”, Trevor Friedman, Robert Tibshirani, Jerome Hastie, Data Mining Information and Prediction – Springer, 2001.
3. "Practical Data Science with R", Nina Zumel and John Mount, Manning Publications, 2014.

Reference Book(s):

1. "R for Data Science", Dan Toomey, PACKT Publishing, 2014.
2. "Visual Explanations – Images and quantities, Evidence and Narrative”, Tufte, Cheshrit, CT: Graphics press 1997
3. "Learning Predictive Analytics with R", Eric Mayor, PACKT Publishing, 2015.
4. “Introduction to time series Analysis”, Peter A Brockwell and Richard A Davis, Springer Texts in statistics, 3rd edition, 2016.
5. “Business Analytics – The Science of Data – Driven Decision Making”, Dinesh Kumar, Wiley India, 2017.

UE17CS323:
FUZZY LOGIC (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Learn about formal methods to represent “vague” and “less” mathematical knowledge.
- Combine some of the traditional design approaches with Fuzzy Logic.
- Exposure to new and exciting applications of “vague” knowledge processing.

Course Outcomes:
At the end of the course, the student will be able to:
- Have a broad knowledge of Fuzzy Logic operations.
- Demonstrate the ability to think critically in making decisions based on Fuzzy Logic.
- Apply a new thinking methodology to real life problems including engineering ones.

Course Content:
1. **Introduction, Classical Sets and Fuzzy Sets:** The Case for Imprecision, A Historical Perspective, The Utility of Fuzzy Systems, Limitations of Fuzzy Systems, The Illusion: Uncertainty and Accuracy, Uncertainty and Information, Fuzzy Sets and Membership, Chance versus Fuzziness, Sets as Points in Hypercube Classical Sets - Operations on Classical Sets, Properties of Classical (Crisp) Sets, Mapping of Classical Sets to Functions, Fuzzy Sets - Fuzzy Set Operations, Properties of Fuzzy Sets, Alternative Fuzzy Set Operations.
2. **Classical Relations and Fuzzy Relations:** Cartesian Product, Crisp Relations - Cardinality of Crisp Relations, Operations on Crisp Relations, Properties of Crisp Relations, Composition, Fuzzy Relations - Cardinality of Fuzzy Relations, Operations on Fuzzy Relations, Properties of Fuzzy Relations, Fuzzy Cartesian Product and Composition, Non - Interactive Fuzzy Sets, Tolerance and Equivalence Relations - Crisp Equivalence Relation, Crisp Tolerance Relation, Fuzzy Tolerance and Equivalence Relations, Value Assignments - Cosine Amplitude, Max-Min Method, Other Similarity Methods.
3. **Properties of Membership Functions, Fuzzification and Defuzzification:** Features of the Membership Function, Various Forms, Fuzzification, Defuzzification to Crisp Sets, Lambda -Cuts for Fuzzy Sets, Lambda - Cuts for Fuzzy Relations, Defuzzification to Scalars, Development of Membership Functions - Membership Value Assignments (Intuition, Inference, Rank Ordering, Neural Networks, Genetic Algorithms), Fuzzy Arithmetic and Extension Principle: Extension Principle - Crisp Functions, Mapping and Relations, Functions of Fuzzy Sets – Extension Principle, Fuzzy Transform (Mapping), Practical Considerations, Fuzzy Arithmetic, Interval Analysis in Arithmetic, Approximate Methods of Extension - Vertex Method, DSW Algorithm, Restricted DSW Algorithm, Comparisons.
4. **Logic and Fuzzy Systems:** Classical Predicate Logic – Tautologies, Contradictions, Equivalence, Exclusive OR and Neither Exclusive NOR, Logical Proofs, Deductive Inferences, Fuzzy Logic, Approximate Reasoning, Other forms of the Implication Operation. **Fuzzy Systems:** Natural Language, Linguistic Hedges, Rule Based Systems Multiple Conjunctive Antecedents, Multiple Disjunctive Antecedents, Aggregation of Fuzzy Rules, Graphical Techniques of Inference.
5. **Decision Making with Fuzzy Information:** Fuzzy Synthetic Evaluation, Fuzzy Ordering, Non - Transitive Ranking, Preference and Consensus, Multi Objective Decision Making, Fuzzy Bayesian Decision Method, Decision Making under Fuzzy States and Fuzzy Actions. **Fuzzy Classification:** Classification by Equivalence Relations - Crisp Relations, Fuzzy Relations, Cluster Analysis, Cluster Validity, c-

Means Clustering - Hard c-Means (HCM), Fuzzy c-Means (FCM), Classification Metric, Hardening the Fuzzy c-Partition, Similarity Relations from Clustering.

Pre-requisite Courses: UE17CS205 - Discrete Mathematics and Logic.

Reference Book(s):

1. "Fuzzy Logic with Engineering Applications", Timothy J. Ross, Wiley India, 3rd Edition, 2010.
2. "Fuzzy Sets and Fuzzy Logic: Theory and Applications", George K Klirand and Bo Yuan, Prentice Hall, 1995.
3. "Neural Networks and Fuzzy Systems: A Dynamical System Approach", B Kosko, PHI, 1991.

UE17CS324:
COMPUTATIONAL APPROXIMATION – METHODS AND ALGORITHMS (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Explore complex systems, physicists, engineers, financiers and mathematicians
- Familiarize with complexity and stability theory
- Expose mathematical and computational foundations of numerical approximation
- Become conversant with solution of large scale systems of linear and nonlinear equations

Course Outcomes:

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

- Be able to understand and implement basic algorithms underpinning computer predictions in modern systems science
- Have a good understanding of the numerical approaches to problem solving
- Use friendly and informal mathematical treatment to represent system approximation
- Adapt to efficient algorithmic approaches for computational approximation
- Analyze and evaluate the accuracy of common numerical methods implemented in MATLab

Course Content:

1. Introduction & Basics: Vectorisation. Polynomial interpolation, precision, floating points, Numerical differentiation and integration
2. Splines: Linear and cubic splines in one dimension. Radial basis function splines in multiple dimensions
3. Matrices and Systems: LU and QR factorisation and applications Norms and condition numbers. Jacobi method; fixed point iteration, Newton's method. Euler's method, Improved Euler method, Initial-value problems
4. Approximating systems of Differential Equations: Runge Kutta methods, time-step limitations, Matlab ODE solvers. Boundary-value problems. Partial differential equations
5. Simulation: Monte Carlo methods, Existence and uniqueness, Banach fixed point, contraction mapping, application in neural networks

Pre-requisite Courses: UE17CS203 – Introduction to Data Science, UE17MA251 – Linear Algebra.

Text Book(s):

1. Numerical Analysis 10th Edition, Richard L. Burden, J. Douglas Faires, Annette M. Burden, Cengage Learning; (Indian Edition), 2015.

Reference Book(s):

1. Differential Equations: A Structured Approach; Snehanshu Saha, Cognella, 2nd Edition, 2011.
2. D. P. O'Leary, Scientific computing with case studies, SIAM, 2008.

UE17CS325: ARTIFICIAL INTELLIGENCE (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Provide an introduction to Machine Intelligence, Problem Solving, Heuristic Search.
- Provide an introduction to Game Playing.
- Provide an introduction to various knowledge representation techniques, reasoning, and expert systems.
- Provide an introduction to planning and learning in AI.
- Introduce Understanding, Natural Language Processing, and Robotics - Perception and Action.

Course Outcomes:

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

- Apply various search techniques for solving problems in AI.
- Write programs to play games.
- Apply knowledge representation techniques and build algorithms for reasoning with knowledge.
- Apply planning and learning algorithms to enhance AI problem solving.
- Identify the AI research and problem areas and choose appropriate problem solving methods.

Course Content:

1. **Introduction to Artificial Intelligence:** Origins, Historical Perspective, Successes and Failures. **State Space Search:** AI Problems, Intelligent Agents, State Space Search, Heuristic Search Techniques.
2. **Advanced Search Techniques:** Game Playing - Adversarial Search, Simulated Annealing, Beam Search, Genetic Algorithm, Constraint Satisfaction Problems.
3. **Knowledge Representation using Logic:** Propositional Logic, First Order Predicate Logic, Reasoning, Logic Programming.
4. **Structured KR Techniques:** Semantic Nets and Frames. **Expert Systems:** Rules, Rule Chaining, Rule Based Systems, Rete Algorithm.

5. **Planning:** STRIPS Model, Goal Stack Planning, Hierarchical Planning, Graph Plan. **Introduction to Advanced Topics:** Perception and Action, Learning, Understanding and NLP. **Current Applications in AI**: Robotics, Gaming.

Pre-requisite Courses: UE17CS251 – Design and Analysis of Algorithms.

Reference Book(s):

1. "A First Course in Artificial Intelligence", Deepak Khemani, McGraw Hill, 1st Edition, 2013.
2. "Artificial Intelligence – A Modern Approach", Stuart Russell and Peter Norvig, Pearson, 3rd Edition (Paperback), 2016.
3. "Artificial Intelligence", E Rich and K Knight and Shivashankar Nair, Tata McGraw Hill, 3rd Edition, 2009.

UE17CS351:
COMPILER DESIGN (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- 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 a compiler's perspective.

Course Outcomes:

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

- Use the knowledge of patterns, tokens and regex for solving the problems in the field of Data Mining.
- Analyze and design the semantic behaviour of a compiler.
- Design and develop the behaviour of a construct.
- Design and implement a simple compiler.
- Optimize the performance of a program in terms of speed and space using new code optimization techniques.

Course Content:

1. **Compilers:** The Language Processing System, The Phases of a Compiler, The Grouping of Phases into Passes, Variations in Compiler Models**. Lexical Analysis:** The Role of the Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, Design of a Lexical Analyzer Generator. **Syntax Analysis:** The role of Parser, Syntax Error Handling, Error-Recovery Strategies. **Top - Down Parsing:** Recursive Descent Parser (RDP) with Backtracking.
2. **Syntax Analysis: Top - Down Parsing:** LL (1) Parser. **Bottom – Up Parsing:** Shift - Reduce Parsing, LR(0), SLR, Viable Prefixes, CLR, LALR.
3. **Syntax – Directed Translation:** Syntax - Directed Definitions (SDDs), Evaluation Orders for SDDs, Applications of Syntax - Directed Translation (SDT), Syntax - Directed Translation Schemes – Postfix Translation Schemes. **Parser**

Stack Implementation: Parser Stack Implementation of Postfix SDTs, SDTs with Actions Inside Productions, SDTs for L - Attributed Definitions.

4. **Syntax - Directed Translation - Implementing L - Attributed SDDs**: Bottom - Up Parsing. **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.
5. **Machine Independent Optimization:** Optimization of Basic Blocks**,** Live-Variable Analysis**. Run - Time Environments:** Storage Organization, 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

Pre-requisite Courses: UE17CS202 – Data Structures, UE17CS254 – Theory of Computation.

Reference Book(s):

1. "Compilers – Principles, Techniques and Tools", Alfred V Aho, Monica S Lam, Ravi Sethi, Jeffery D Ullman, Pearson Education, 2nd Edition, 2009.
2. "Modern Compiler Design", Dick Grune, Kees van Reeuwijk, Henri E Bal, Ceriel J.H. Jacobs, Koen Langendoen, 2nd Edition, 2012.

UE17CS352:
CLOUD COMPUTING (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Understand the rationale behind the cloud computing revolution
- Introduce various models of cloud computing
- Understand how to design applications on cloud and the role of security
- Understand and design distributed systems for scalability
- Understand and apply various tradeoffs in designing cloud architectures

Course Outcomes:

At the end of the 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
- Apply the concepts learnt in solving a real life problem in a group setting

Course Content:

1. **Introduction and Cloud models:** Cloud computing – business perspective, models SOA and REST, IaaS model and case study, PaaS Model
2. **Virtualization Compute -** CPU virtualization – hardware/software - introduction, transparent and para virtualization, Lightweight virtualization – containers - Docker. Microservices,
3. **Storage and Network Virtualization -** CAP Theorem – Distributed storage, Object and Block storage virtualization, Shared storage, Scaling Storage, Multitenancy and SaaS Model, Introduction to network virtualization

4. **Orchestration and DevOps:** Algorithms for allocation of compute, storage. Container Orchestration - kubernetes, Continuous development – DevOps-components and tools
5. **Distributed architectures:** Architectures – peer to peer, master slave, cluster coordination, failure – availability/reliability, Cluster Coordination, Case study: zookeeper, Security in cloud

Pre-requisite Courses: None.

Reference Book(s):

1. "Moving to the clouds: Developing Apps in the new world of cloud computing", Dinkar Sitaram and Geetha Manjunath, Syngress, 2011.
2. Microservice Architecture Aligning Principles, Practices, and Culture, O'Reilly, 2016
3. "Learning Openstack", Alok Shrivsastwa and Sunil Sarat, PACKT Publishing, 2015
4. Cloud Native DevOps with Kubernetes – Building, Deploying and Scaling Modern Applications in the Cloud, John Arundel and Justin Domingus, O'Reilly, 2019

UE17CS353:
WEB TECHNOLOGIES II (4-0-0-0-4)

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

- Provide an understanding of designing advanced Web Applications using AJAX techniques.
- Provide an introduction to challenges involved in improving performance of rich web sites.
- Learn the basics of Web Services and Reverse AJAX techniques.
- Provide an overview of various Web Frameworks.
- Provide a basic overview of non-functional side of the WWW.

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

- Design Advanced Web Applications, with techniques like AJAX, Server-Push, etc.
- Use Node.js as a Server-Side Framework to develop web sites that provide fast and reliable content.
- Develop RESTful Web Services and consume them.
- Develop robust web sites that are immune to malicious web attacks.

Course Content:

1. **AJAX:** JS objects, prototype inheritance, Ajax **-** Hidden Frames Technique, Image-Based AJAX, Dynamic Script Loading, XMLHttpRequest, Cross-Domain Access (CORS), Maintaining History in AJAX Calls, Fetching Binary Data with XHR
2. **AJAX Patterns and Data Formats** – Predictive Fetch, Multi-Stage download, Periodic Refresh and Fallback Patterns, Submission Throttling. Introduction to XML, Parsers, Styling RSS / Atom Feeds, JSON and XML, JSON vs XML.
3. **Reverse AJAX:** COMET, HTTP Streaming and Long Poll, iFrames Technique, Using XMLHttpRequest, HTML5 Server Sent Events, Overview of COMET Frameworks,

Web Sockets. **Web Services**: Principles of REST, Architecture of SOAP-Based Services, REST vs. SOAP Based Services, Building and Invoking RESTful Web Services using PHP, Introduction to Micro Services.

4. **Non-Functional Aspects of the Web:** Performance Considerations - Timeouts, Retries, Handling Server Errors, Multiple Requests, The HTTP 1.1 Two Connection Limit, Race Conditions, Caching on Client Side, Compression of Data, Request Management, Scalability. HTTP 2.0 – New Features, HTTP 2.0 vs. 1.1. Security – Web Attack Surfaces, Reconnaissance Review, Various Vulnerabilities and Precautions, SQL Injection, XSS, XST, CSRF, HTTP Authentication and its Configuration in Apache.
5. **Web Frameworks:** Overview of Popular Frameworks. **Node.js**: Basics, Callbacks, HTTP Responses, File and Database Access. **Angular JS:** Introduction, Directives and Controllers, Forms, Inputs and Services, Server Communication with $http, Filters.

Pre-requisite Courses: UE17CS204 – Web Technologies I.

Reference Book(s):

1. "Professional AJAX", Nicholas C. Zakas et. al, Wiley Publishing, 2nd Edition, 2007.
2. "Angular JS Up and Running", Shyam Sheshadri and Brad Green, O Reily, 1st Edition, 2014.
3. "AJAX: The Complete Reference", Thomas A Powell, McGraw Hill, 2008.
4. Handouts for Web Sockets, HTTP 2.0 protocol, Node.js.

UE17CS354:
COMPILER DESIGN LABORATORY (0-0-2-0-1)

Course Objectives:

The objective(s) of this course is to,

- Deepen the understanding of compiler design
- Develop problem solving ability using programming
- Develop ability to design and analyze a compiler

Course Outcomes:

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

- Acquire the generic skills to design and implement a compiler
- Analyze practical aspects.

Course Content:

1. Lex to count number of char, words, newlines, and white spaces
2. Lex to remove comment lines
3. Yacc to validate identifiers; Yacc to validate context free grammars
4. Yacc to implement semantic rules to calculate the expression that takes an expression with digits, + and * and computes the value.
5. Yacc to validate Nested IF–ELSE construct. Check whether a given C program is syntactically correct
6. Yacc to generate abstract syntax tree for a given C program.
7. Week 7 – 13 : Compiler Design Mini Project

Pre-requisite Courses: None.

Reference Book(s):

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

UE17CS355:
WEB-TECHNOLOGIES II LABORATORY (0-0-2-0-1)

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

- Learn and implement asynchronous communication techniques using AJAX.
- Learn and implement AJAX Design Patterns
- Learn and implement Rich Front End using AngularJS
- Learn to implement REST-based web services with XML and JSON.
- Learn to implement web servers using Node.js

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

- Build Rich and efficient Web Applications using AJAX techniques
- Implement and offer RESTful Web Services with XML and JSON
- Deploy Web Applications on servers with Node.js as the back-end
- Deploy reverse AJAX techniques in Web Applications
- Understand Web Attacks and common defense tactics.

Course Content:

1. Introduction to Lab Environment and Basic Instructions
2. Mini Project Team and Title finalization.
3. Demonstrate the hidden frames AJAX technique (both GET and POST)
4. Demonstrate Video streaming using XMLHttpRequest
5. Demonstrate Multistage download using AJAX
6. Demonstrate Periodic refresh (with exponential back-off feature).
7. Mini project design and implementation
8. Demonstrate use of AJAX with RSS for automatic updates to web sites
9. Demonstrate usage of Reverse AJAX/Comet
10. Implement and host RESTful web services with both JSON and XML.
11. Demonstrate an XSS attack (non-persistent and persistent)
12. Mini Project Implementation
13. Mini Project Evaluation

Pre-requisite Courses: None.

Reference Book(s):

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

UE17CS331:
COMPUTER NETWORK SECURITY (4-0-0-0-4)

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

- 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

Course Outcomes:
At the end of the course, the student will be able to:
- Fetch or secure the network user's information
- Perform various network attacks and their mitigation strategies
- Aware of the responsibilities of security management
- Assess the risks involved in an organization

Course Content:
1. **Network Security Analysis:** Computer Security Concepts, Requirements, Architecture, Trends, Strategy. Perimeter Security: Firewalls, Intrusion Detection, Intrusion Prevention Systems, Honeypots. Analysis of TCP SYN flood DoS attack. Network packet Analysis: Usage of Banner grabbing tools like nmap, Wireshark. Man-in-the-Middle attacks
2. **Authentication and Access Control:** User Authentication: Password, Password - Based, Token - Based, Biometric, Remote User Authentication. Access Control: Principles, Access Rights, Discretionary Access Control, UNIX File Access Control, Role Based Access Control. Internet Authentication Applications: Kerberos, X.509, PKI, Federated Identity Management. Case study: Access control list for PES University.
3. **Human Factors:** Security Awareness, Training and Education, Organization Security Policy, Employment Practices and Policy. IT Security Management and Risk Assessment: IT Security Management, Risk Assessment, Analysis of IT Security Controls. Plans and Procedures: IT Security Management Implementation, Security Controls, Plan, Implementation of Controls. Case Study: iPremier
4. **Cloud 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.
5. **Wireless Network Security:** 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.

Pre-requisite Courses: UE17CS301 – Computer Networks.

Reference Book(s):

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

UE17CS332:
STORAGE AREA NETWORKS (4-0-0-0-4)

Course Objectives:
The objective(s) of this course is to,
- Introduce students to the current storage technologies.
- Equip students with an understanding of SAN technologies in Data Centers, SAN architectures, its advantages and complexities.

- Walk through the IO path from an Application to bits and bytes stored in a device with all the related technologies in the path.
- Provide hands on exposure to Cloud Storage and how you would write/ read data there.
- Exposure to SAN applications like Backup, Security and Management aspects of SAN/ NAS.

Course Outcomes:
At the end of the course, the student will be able to:
- Analyze the logical and physical components of a storage infrastructure.
- Design different types of RAID implementations and enumerate their benefits.
- Analyze and walk through the data path from Application to a Disk
- Contrast SCSI, Fiber Channel and iSCSI protocols.
- Analyze benefits of storage virtualization.
- Write an application which would write/ read data from a Cloud Storage.

Course Content:
1. **Introduction to Storage Systems, IO Techniques and Intelligent Disk Systems:** Structured and Unstructured Data, Data Centers, Key Requirements of Data Centers, Storage in Data Centers, Types of Storage (including SDS), Converged Storage, Futures in Storage Technologies, Storage – Centric IT Architecture and Its Advantages, The Physical I/O Path from the CPU to the Storage System. **Introduction to Storage Components:** Disks, Physical Structure of Disks, Categorization of Disk Subsystems, Architecture of a Disk Subsystem, Internal I/O Channels and Their Design Variations, RAID Levels, Availability of Disk Systems.
2. **Storage Protocols - SCSI, FC, iSCSI:** SCSI Basics, Components, Addressing, Protocol and Functioning. **Fiber Channel:** FC Basics, FC Protocol Stack, iSCSI.
3. **SAN and NAS – Components and Hardware and Software Architecture:** Network Attached Storage - Introduction NAS Hardware and Software Components and Architecture, SAN Hardware and Software Components and Architecture.
4. **Storage Virtualization and Large Storage Systems:** Storage Virtualization, Implementation Considerations, Server, Device and Network Centric Storage Virtualization, Storage Virtualization in Block or File Level, In - Band, Out-Band, IP Storage, Object Storage, Cloud Storage. **Illustrations of Cloud Storage:** Google FS, Openstack, Amazon S3, MS Azure.
5. **Storage Management and Applications:** SAN and NAS Management, CIM/ WEBEM, SMI-S. **Storage Security:** Security Considerations in Storage, Information Security, Security Methods, Storage Security Technologies and Challenges, Best Practices in Security. **Business Continuity and Storage Applications:** Backup, Recovery and Archival.

Pre-requisite Courses: None.

Reference Book(s):

1. “Storage Networks Explained”, Ulf Troppens, Rainer Erkens and Wolfgang Muller, Wiley India, 2013.
2. “Storage Networking Fundamentals – An Introduction to Storage Devices, Subsystems, Applications, Management, and File Systems”, Marc Farley, Cisco Press, 2005.

3. “Storage Networks - The Complete Reference”, Robert Spalding, Tata McGraw-Hill, 2011.
4. “Storage Area Network Essentials - A Complete Guide to Understanding and Implementing SANs”, Richard Barker and Paul Massiglia, Wiley India, 2006.

UE17CS333:
NATURAL LANGUAGE PROCESSING (4-0-0-0-4)

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

- Expose students to the concepts of n-grams and Language Modelling with n-gram.
- Expose students to the Natural Language Processing pipeline i.e. Morphology, Lexical Analysis, Syntactic Analysis, Semantic Analysis and Discourse.
- Expose students to the Information Extraction problems and end to end Natural Language Generation problems as applications of Natural Language Processing.
- Introduce students to the various Neural Network methods for Natural Language Processing.

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

- Implement meaningful course or research projects using current Natural Language Processing technology.

Course Content:

1. **Text Normalization, Morphology and Finite State Transducer:** Concept/ Types of Ambiguity in Natural Language Processing .Introduction of NLTK for text processing, Edit Distance : Minimum Edit Distance Algorithm, Improved Edit Models.**Text Normalization:**Content and Function Words, Type vs. Token, Unix Tools for Tokenization and Normalization, Word Tokenization and Normalization, Lemmatization and Stemming, Sentence Segmentation.**Morphology and Finite State Transducers:** Survey of English Morphology, Finite State Morphological Parsing, Combining FST Lexicon and Rules, Lexicon - Free FST - The Porter Stemmer.
2. **N-Grams and Language Modelling:** n-grams, Evaluating Language Models - Perplexity, Generalization and Zeros, Smoothing - Kneser-Ney Smoothing, Web and Stupid Back Off, Perplexity’s Relation to Entropy.**Spelling Correction and Noisy Channel:** Noisy Channel Model, Real World Spelling Error, **Word Classes and Part-of-Speech (POS) Tagging:** English Word Classes, Penn Tagsets for English, Rule-Based Part-of-Speech Tagging, Transformation-Based Tagging, POS Tagging using Hidden Markov Model, Maximum Entropy Model and Conditional Random Fields.
3. **Parsing:** Context Free Grammar. **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 PCFG. **Introduction to Dependency Parsing:** Dependency Relations, Dependency Formalisms, Dependency Tree Banks, Evaluating Parsers.
4. **Semantics - Lexical semantics:** Word Senses and Relations Between Word Senses, WordNet: A Database of Lexical Relations, Word Sense Disambiguation - Overview, Supervised Word Sense Disambiguation, WSD - Dictionary and

Thesaurus Methods, Semi-Supervised WSD, Unsupervised Word Sense Induction.**Word Similarity or Semantic Relatedness Based On Thesaurus:** Resnik Similarity, Lin Similarity, Jiang-Conrath Distance, Extended Gloss Overlap And Extended Lesk Method.**Lexicons For Sentiment and Affect Extraction:** Available Sentiment Lexicons, Using Wordnet Synonyms And Antonyms - Sentiwordnet, Supervised Learning of Word Sentiments, Using Lexicon For Sentiment Recognition, Lexicons For Emotions And Other Affective States.**Representation Of Meaning:** Computational Desiderata for Representations, Meaning Structure of Language, First Order Predicate Calculus, Some Linguistically Relevant Concepts, Related Representational Approaches, Alternative Approaches To Meaning. Introduction to regional language processing. **Co-Reference Resolution and Discourse:** Lappin and Lease Algorithm, Hobbs Algorithm.

5. **Information Retrieval, Natural Language Generation and Neural Network Methods for Natural Language Processing - Information retrieval:** Information Extraction vs. Retrieval, Information Extraction Sub-Problems, Named Entity Recognition - Practical NER Architectures. **Natural Language Generation:** An Architecture, Question Answering System - IR Based Factoid Question Answering, Knowledge Based Question Answering, IBM's Watson, Dialogue System And Chatbot - Rule Based And Corpus Based Chatbots.**Vector Semantics:** Words And Vectors, Pointwise Mutual Information, Measuring Similarity, Using Syntax to Define a Word's Context, Evaluating Vector Models.**Semantics With Dense Vectors:** Dense Vectors via SVD, Distributional Hypothesis, Embedding from Predictions - Skip Gram And CBOW, Properties Of Embedding, Pre-Trained Word Representations - Word2vec And Glove, Limitation Of Distributional Methods.**Neural Network Methods For Natural Language Processing:** CNN and RNN as Feature Extractors, Modeling With Recurrent Neural Network - Sentiment Classification, Part of Speech Tagging, Conditioned Generation By RNN - Encoder-Decoder, Seq2seq Models, Seq2seq Chatbots.

Pre-requisite Courses: UE17CS251 – Design and Analysis of Algorithms.

Reference Book(s):

1. "Neural Network Methods for Natural Language Processing", Yoav Goldberg , Morgan and Claypool Publishers.
2. "Speech and Natural Language Processing", Daniel Jurafsky and James H. Martin, 3rd Edition Draft, http://web.stanford.edu/~jurafsky/slp3/
3. "Natural Language Processing with Python – Analyzing text with natural Language Toolkit", Steven Bird, Ewan Klein, and Edward Loper, O'rielly books, https://www.nltk.org/book/
4. "Deep Learning with Python", Francois Chollet, Manning.

UE17CS334:
HIGH PERFORMANCE COMPUTING ARCHITECTURE (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Introduce the factors influencing a program's performance.
- Introduce concepts from processor architecture, compiler optimizations and software approaches.
- Introduce various parallel architectures like multi-core and GPUs.

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

- Analyze a given program and measure its performance based on various factors.
- Write and debug performance efficient code.
- Write task and data parallel programs.

Course Content:

1. **Understanding Performance, Factors Influencing Performance:** Modern Architectures – Instruction Level Parallelism, Multi-Core Architectures, Hyper Threading, Massively Parallel GPU Architectures. **Measuring Performance:** Tools to Measure Performance of Programs, Methodology of Measuring and Debugging Performance.
2. **Compiler Optimizations to Achieve Performance:** Understanding Program Structures, Effect on Compilers, Loop Optimizations – Unrolling, Fusion, Transformations, Memory Hierarchy and Effect on Performance, Cache Organizations, Writing Cache Friendly Code.
3. **Threads:** Creation/ Deletion/ Scheduling of Threads, Synchronization of Threads, Deadlocks. **Parallel Programming Models:** Task Level and Data Level Parallelism.
4. **Introduction to OpenMP:** For Extracting Parallelism on Modern Processors.
5. **Introduction to OpenCL/ CUDA**: For Extracting Parallelism on Modern GPUs.

Pre-requisite Courses: UE17CS253 – Microprocessor and Computer Architecture.

Reference Book(s):

1. "Computer Architecture: A Quantitative Approach", Hennessey and Patterson, Morgan Kaufmann, 2011.
2. "Computer Systems: A Programmer's Perspective", Randal Bryant and David O'Halloran, Prentice Hall, 2nd Edition, 2011.
3. "Programming Massively Parallel Processors: A Hands on Approach", David Kirk and Wen-mei Hwu, Morgan Kaufmann, 2010.
4. "Theory of Computation: A Problem-Solving Approach", Kavi Mahesh, Wiley India, New Delhi, 2012.
5. "Introduction to Automata Theory, Languages and Computation", John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education India, 3rd Edition, 2009.
6. "Theory of Computation", Michael Sipser, Cengage Learning, New Delhi, India, 2008.

UE17CS335:
PARALLEL COMPUTING (4-0-0-0-4)

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

- Understand various models of parallel computations such as threads, OpenMP, MPI, clusters.
- Develop programs using parallel languages and appreciate parallel compilers using parallel architectures.

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

- Design efficient parallel algorithms and applications.

- Analyze the effectiveness of any parallel program.

Course Content:

1. **Parallel Programming Models:** Introduction to Analytical Models and Message Passing Communications (MPI), Review of Shared Memory Programming Models (OpenMP/ CUDA).
2. **Parallel Algorithms:** Sequences and String, Scan (Prefix Sums), List Ranking, Sorting, Merging, Medians, Searching, String Matching.
3. **Graph Algorithms**: Trees and Graphs - Spanning Trees, Shortest Path, Transitive Closure, Connected Components.
4. **Computational Geometry:** Convex Hull, Closest Pairs. **Numeric/ Scientific Applications:** Matrix Multiplication, Sparse Matrix, Map Reduce.
5. **Constructs in Parallel Languages:** Haskell, Scala, GO Lang, C++ 11 and Java 1.8 Features, Concurrent Data Structures - Java 1.8 Concurrent Collections.

Pre-requisite Courses: UE17CS251 – Design and Analysis of Algorithms.

Reference Book(s):

1. "Introduction to Parallel Computing", Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Addison-Wesley, 2003.
2. "Art of Concurrency", Clay Breshears, O'Reilly, 2009.
3. "Parallel Programming in C with MPI and OpenMP", Michael Quinn, TMH, 2003.
4. "Introduction to Parallel Computing", Petersen and Arbenz, Oxford University Press, 2004.
5. "Parallel Programming", Wilkinson and Allen, Pearson, 2006.
6. "Parallel Computer Architecture", Culler, Singh and Gupta, Morgan Kaufmann, 1998.
7. "Parallel Computing – Theory and Practice", Michael Quinn, McGraw-Hill, 1993.
8. "Introduction to Automata Theory, Languages, and Computation", John E Hopcroft, Rajeev Motwani, Jeffrey D Ullman, Pearson Education India, 3rd Edition, 2009.

UE17CS336:
GENERIC PROGRAMMING (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Understand the rationale behind generic programming.
- Understand compile time checks.
- Gain insights into the concept of meta programming.

Course Outcomes:

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

- Solve programming problems and improve programs using Generic Programming.
- Evaluate Generic Programming methods/ approaches and understand their pros and cons.

Course Content:

1. **Template Functions:** Definition, Instantiation - Implicit and Explicit, Specialization, Type and Non-Type Template Parameter.
2. **Template Class:** Instantiation, Templates and Static Members, Templates and Inheritance, Templates and Composition, Templates and Friends, Template Member Functions, Dependent Type, Default Template Parameter, Nested Templates Traits and Policies, STL Philosophy, Efficiency of Algorithms, Separation of Behaviour from Container Classes, Functor and Iterator, Iterator Hierarchy, Adaptors, Examples of Containers and Algorithms.
3. **Template Meta - Programming Overview:** Compile-Time Programming Nature and Limitations of Template Meta-Programming.
4. **Building Blocks:** Values, Functions, Branching, Recursion, Compile-Time "If" Conventions for "Structured" Template Meta Programming.
5. **Generics in Java:** Generic Methods, Constructors, Type Inference, Bounded Type Parameters, 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.

Pre-requisite Courses: None.

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. “STL Tutorial and Reference”, Musser, Derge and Saini, 2nd Edition, Addison-Wesley, 2001.
5. “Java Tutorials”, Online Reference Link - https://docs.oracle.com/javase/tutorial/.
6. MSDN for C# generics.

UE17CS337:
DRONE COMPUTING (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- To enable the students to have hands-on experience with drones/quadcopters.
- To understand the basic principles of flying.
- To write program to fly drones.
- To use mission planners for accomplishing GPS related tasks.
- To enrich the drone computing applications to serve the society.

Course Outcomes:

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

- Fly drones/ quadcopter both using RF controller and semi-autonomously
- Write programs to guide the paths of drones
- Use GPS coordinates to locate positions while flying
- Apply the fundaments of drone computing for their projects
- Innovate workable ideas to address the challenging problems in the society

Course Content:

1. **Building a Simple Drone:** Introduction to Drones, Types of Drones, Configuration of Flight Controller, Configuration of Receiver Transmitter Control, Assembling Drone Components, Experiments with Simple Flight Paths.
2. **Understanding Control:** Theory of PTD Controller, Experiments Flying with PID, Theory of Electronic Control, Experiments Flying with EC, Analyzing Path Accuracies and Flight, Stability with Experiments.
3. **Path Planning with GPS:** Introduction to GPS, Experiments with GPS Modules, Path Planning with GPS Co-ordinates, Experimenting Path Planning with GPS Co-ordinates, Analyzing Path Accuracies and Flight Stability with Experiments.
4. **Impact of Payload:** Introduction to Payload, Impact of Payload, Maximizing Payload, Experiments with Different Payloads, Analyzing Path Accuracies and Flight Stability with Experiments.
5. **Tracking with Camera:** Introduction to Tracking with Camera, Object Recognition, Obstacle Avoidance, Experiments for Tracking Objects, Analyzing Follow-Up Accuracies and Flight Stability with Experiments.

Pre-requisite Courses: None.

Reference Book(s):

1. "Getting Started with Drones: Build and Customize Your Own Quadcopter", Terry Kilby, Belinda Kilby, Maker Media, Inc, 1st Edition, 6 October 2015.
2. "Make: Drones: Teach an Arduino to Fly", David McGriffy, Maker Media, Inc, 1st Edition, 10 October 2016.
3. "Drone/ UAV Dictionary: Includes 300 Commercial UAV Applications", Dr. Jerry LeMieux, CreateSpace Independent Publishing Platform, 1st Edition, May 9, 2014.

UE17CS338:
TOPICS IN DEEP LEARNING (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- 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 the course, the student will be able to:

- Implement Machine Learning techniques with TensorFlow and Keras.
- 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.
- Develop simple game engines using Reinforcement Learning.

Course Content:

1. **TensorFlow and Keras and Reinforcement Learning**: Brief overview of Deep Learning 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.

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
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 Naïve Bayes, Unsupervised Feature Learning – Autoencoders and Variations.
4. **CNN, GAN and Transfer Learning: CNN -** Architecture of CNNs, Filters, Feature Maps, Max-Pool Layers, Other Pooling Types, Case Study: Image Recognition Using CNN – Hands-On Implementation Using Keras. **GAN** - Architecture and Training Methods, Image-Generation, Hands-On Implementation Using Keras. **Transfer Learning** - Motivation, Variations, Use in CNNs.
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.

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

Pre-requisite Courses: UE16CS353 – Machine Learning.

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.

UE17CS341:
SOFTWARE DEFINED NETWORKS (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Understand the rationale behind evolution of Software Defined Network technology.
- Understand SDN based managed network services through abstraction of higher-level functionality.
- Understand the architecture of SDN traffic through Control and Data planes.

Course Outcomes:

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

- Implement SDN approach to design, engineer and manage the networks.
- Configure Network services using SDN with ease, efficiently and remotely monitor QoS.

Course Content:

1. History and evolution of Software Defined Network, SDN Architecture.
2. Control and Data Plane Separation, Control Plane, Network Virtualization, Data Plane.
3. Programming SDNs, Openflow.
4. Verification and Debugging, Use Cases and Looking Forward.
5. Hands-on using simulator Mininet.

Pre-requisite Courses: UE17CS301 – Computer Networks.

Reference Book(s):

1. "Software Defined Networks - Programmability Technologies", Thomas D. Nadeau, Ken Gray -SDN: O'Reilly Media, 2013.
2. "Software Defined Networks - A Comprehensive Approach", Paul Goransson and Chuck Black, Morgan Kaufmann, 2014.

UE17CS342:
KNOWLEDGE MANAGEMENT (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Understand and appreciate the rationale and theory of Knowledge Management.
- Learn various tools for Knowledge Management.
- Understand the significance of Knowledge Management.
- Learn different ways of knowledge creation and transfer.
- Understand the role and implications of technology in the field of Knowledge Management.

Course Outcomes:

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

- Apply Knowledge Management paradigms in applications like Semantic Web.
- Select and use various tools for effective Knowledge Management.

Course Content:

1. **The Basics:** Working Smarter, Knowledge Management (KM) Myths and Lifecycle, Implications of KM. **Understanding Knowledge**: Definitions, Cognition and KM, Data, Information and Knowledge, Types of Knowledge, Expert Knowledge, Human Thinking and Learning, Implications for KM. **Knowledge Management Systems Lifecycle**: Challenges, Conventional versus KM System Lifecycle, Implications for KM. **KM Strategy**: Economy of Plan, Economy of Change, Economy of Control.
2. **Knowledge Creation and Capture:** Knowledge Creation and Knowledge Architecture, Nonaka's Model, Knowledge Architecture, Implications, Capturing Tacit Knowledge, Knowledge Capture, Evaluating the Expert, Developing a Relationship with Experts, Fuzzy Reasoning and Quality of Knowledge, Interview as a Tool, Guide to a Successful Interview, Rapid Prototyping, Implications.
3. **Design of KM Systems:** Economy of Scope, Economy of Effort, Economy in

Deployment, Other Knowledge Capturing Techniques, Onsite Observation, Brain-Storming, Protocol Analysis, Consensus Decision Making, The Repertory Grid, NGT, Delphi Method, Concept Mapping and Black-Boarding, Knowledge Codification, Why Codify, Modes of Knowledge Conversion, How to Codify Knowledge, Codification Tools and Procedures, Knowledge Developer's Skill Set, Implications. **System Testing and Deployment**: Quality and Assurance, Knowledge Testing, Approaches to Logical and User Acceptance Testing, Managing the Test Phase, KM System Deployment, Issues, User Training and Deployment, Post-Implementation Review, Implications, Knowledge Transfer and Sharing - as a Step in a Process, Transfer Methods, Role of Internet, Implications.

4. **Knowledge Transfer in E-World:** The E-World, E-Business, Implications, KM System Tools and Portals – Learning from Data, Data Visualization, Neural Networks as a Learning Model, Association Rules, Classification Types, Implications, Data Mining – Knowing the Unknown, Data Mining and Business Intelligence, Business and Technical Drivers, Data Mining Business Cycle and Data Management, Data Mining in Practice, Role of DM in Customer Relationship, Implications. **Knowledge Management Tools and Portals:** Portals - The Basics, Business Challenge, Knowledge Portal.
5. **Technologies, Implications, Ethical, Legal and Managerial Issues:** Knowledge Owners, Legal Issues, Ethical Factor, Improving the Climate, Implications.

Pre-requisite Courses: None.

Reference Book(s):

1. "Knowledge Management", Elias M. Awad and Hassan Ghaziri, Pearson Publications, 2007.
2. "Ten Steps to Maturity in Knowledge Management – Lessons in Economy", J.K. Suresh and Kavi Mahesh, Chandos Publishing, 2006.

UE17CS343:
SYSTEM MODELING AND SIMULATION (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- To provide a background to students on the principles of modelling and simulation and expose them to different types of models.
- To train students on the techniques of how to model a scenario using discrete event system models. Expose students to typical applications such as queuing systems and inventory systems.
- To train students to conduct end-to-end simulation experiments and assess the fitness of simulation.
- To expose students to continuous systems and application of continuous simulation to real life systems.
- To expose students to agent-based modelling.

Course Outcomes:

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

- Model a given scenario/ problem and generate inputs to simulate the same and derive simulation table that includes inputs and outputs.
- Apply simulation techniques to typical discrete event systems such as queuing systems, inventory systems and reliability of machines.

- Generate inputs for simulation using understanding of statistical distributions.
- Validate the input models and assess the simulation output relative to real life data.
- Model continuous systems using numeric methods.
- Apply agent-based modelling techniques.

Course Content:

1. **Introduction to Simulation:** Principles of Modelling and Simulation, Model Taxonomies, Fundamentals of Queuing Theory, Random Variate Generation, Monte Carlo Simulation, Performance Measures of Queuing Systems.
2. **Building Discrete Event Simulation Models:** Managing Event Lists, Queue Disciplines, Priorities, Application to Simulation of Computer Subsystems and Concurrent Processes.
3. **Design of Simulation Experiments:** Generation of Random Numbers, Validation of Random Numbers, Generating Variates, Input Data Analysis to Determine Distributions.
4. **Validation of Simulation and Output Analysis**: Verification and Validation of Simulation Models, Analysis of Simulation Output, Tests of Significance and Design of Experiments, Variance Reduction Techniques.
5. **Continuous System Simulation and Agent Based Simulation:** Modelling Systems using Differential Equations, Principles of Numerical Integration, Numerical Integration Methods, Simulation of Discontinuities (Combined Discrete-Continuous Simulation), Application to Population Ecology and Other Systems, The Need for Agent-Based Simulation, Agent Concepts – Characteristics and Interaction Topologies, Agent-Based Simulation Platforms (Netlogo and Others), Applications Examples.

Pre-requisite Courses: UE17CS203 – Introduction to Data Science

Reference Book(s):

1. "Discrete Event System Simulation", Jerry Banks et al, Pearson, , 5th Edition, 2013.
2. "Numerical Methods for Engineers", Steven C Chapra and Raymond P. Canale, 6th Edition, McGraw Hill, 2015.
3. "Simulation with Arena", W David Kelton, 5th Edition, McGraw Hill, 2013.
4. "An Introduction to Agent Based Modelling", Uri Wilenksy , Willam Rand, MIT press, 2015

UE17CS344:
NETWORK MANAGEMENT (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Challenges associated with Network Management (NM).
- Standards related to Network Management.
- Applications related to NM like FCAPS.
- Building of a NM interaction using SNMP.
- Evolving technologies with NM.

Course Outcomes:

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

- Analyze the challenges associated with Network Management.

- Contrast the different NM architectures and their salient points
- Demonstrate applications associated with NM using some of the Open source tools
- Create a MIB, Manager and Agent using SNMP and demonstrate the basics of NM

Course Content:

1. **Introduction to Network Management**: Introduction, What is Network Management, Element and Network Management Systems, Challenges associated with Managing Networks, Variety and Multi-Vendor Environments, Scale and Complexity, Types of Networks, Network Topologies, Functions and Types, Internet And Network Management; Administrative Entities, Their Autonomy and Responsibilities in Terms of Policies, Data Centre Networks. NM – The Big Picture and the Activities/IT Groups which Support them, NOCs, Network Devices and Services which Need to be Managed, Benefits of Automation, Maturity of Network Management, What are the Expectations from an NMS; In an Element based NMS, Who is Best Suited for Providing a Network Management System to Manage the Components? What should a typical NM system provide, Architectural Approaches of NMS.
2. **Standards based Network Management Models**: IETF and OSI, CIM/WEBEM, Interoperability Models. **Network Element Management**: Functional Grouping of Network Management Activities. **A Review of Network Elements and Services:** Introduction, Network Devices and Network Services, Network Elements and Element Management, Effect of Physical Organization on Management, Examples of Network Elements - Hardware components like Switches, Cable Modem System, DSL Modem System and DSLAM, IP Router Software Components like Firewall, DNS Server, DHCP Server, Web Server, HTTP Load Balancer.
3. **FCAPS:** The Industry Standard Definition, Organization of Management Software. **Configuration Management:** Introduction, Intuition for Configuration, Configuration and Protocol Layering, Dependencies among Configuration Parameters, Configuration and Temporal Consequences, Configuration and Global Consistency, Configuration and Default Values, Configuration Types and Ways of Keeping the Environment Stable in Terms of Transactions, Rollbacks etc. **Fault Management:** Fault Management Basics, Detection, Correlation, Isolation, Tracking of Faults and Resolution of Faults, Fault detection tools, Benefits of Fault management tools.
4. **Performance Management:** What is Network Performance Management, Network Performance Measures, How Are They Measured, Where Are They Measured, Capacity Planning, QoS. **Accounting Management**: What is Accounting Management, Context of Billing and Accounting in NM, Account Management Architecture and Process, Service Providers - Revenues and The Relationship to Accounting, SLAs, Billing. **Security Management:** Basics of Security, Network Security, Management Goals Related to Security, Common Security Attacks, Security Policies, Risk Assessment, Technologies for Network Security, Security Architecture.

5. **Introduction to Software Defined Network's Relationship to NM**: Autonomic Computing. **Simple Network Management Protocol (SNMP):** Topics for Writing a MIB, Manager and Agent and Their implementation. **Network Management with BYODs, Network Management with Hybrid Cloud Challenges and the Approach.**

Pre-requisite Courses: UE17CS301 – Computer Networks.

Reference Book(s):

1. "Automated Network Management Systems", Douglas E Comer, 1st Edition, Addison-Wesley, 2012.
2. "Communications Network Management", Terplan Kornel, Prentice Hall, 2nd Edition, 1991.
3. "SNMP, SNMPv2 and CMIP", William Stallings, Addison-Wesley, 1993.
4. "Network Management Principles and Practice", Mani Subramanian, Pearson Education, 2010.
5. "Network Management Fundamentals", Alexander Clemm, CISCO Press, 2007.
6. "A Practical Approach to WEBEM/CIM Management", Chris Hobbs, CRC Press, 2004.
7. "Advances in Network Management", Jianguo Ding, 2010.
8. Various Articles from Internet, Published Papers.

UE17CS345:
DIGITAL IMAGE PROCESSING (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- 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 the 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
- Design and/ or apply algorithms to solve problems pertaining to image processing and analysis

Course Content:

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.
2. **Image Enhancement in the Spatial Domain (continued):** 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.
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.
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.
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 Convolution Neural Networks.

Pre-requisite Courses: UE17CS251 – Design and Analysis of Algorithms.

Text Book:

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

Reference Book(s):

1. Scott.E.Umbaugh: Digital Image Processing and Analysis, CRC Press, 2014.
2. S. Jayaraman, S. Esakkirajan, T. Veerakumar: Digital Image Processing, McGraw Hill Ed. (India) Pvt. Ltd., 2013
3. Tamal Bose: Digital Signal and Image Processing, John Wiley, 2003.

UE17CS346:
ADVANCED COMPUTER NETWORKS (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Prepare students to design and manage various aspects of organizational network.
- Expose students towards real time network design.
- Design network that meet customer’s business and technical goals.
- Provide ability to analyse the existing network design.
- Provide hands-on experience for network design with the help of various simulators.

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

- Design network on predefined network requirements.
- Analyse protocol behaviour, internetworking technologies and traffic flow.
- Select most appropriate routing protocols and addressing scheme for given network design.
- Design campus network and use network management strategies in their design.
- Execute SDLC model for network design.

Course Content:

1. **Analyzing Business Goals and Constraints:** Using a Top-Down Network Design, Methodology, Analyzing Business Goals, Analyzing Business Constraints. Analyzing Technical Goals and Tradeoffs: Scalability, Availability, Network Performance, Security,
   Manageability, Usability, Adaptability, Affordability, Making Network Design Tradeoffs.
2. **Characterizing the Existing Internetwork**: Characterizing the Network Infrastructure, Checking the Health of the Existing Internetwork. Characterizing Network Traffic: Characterizing Traffic Flow, Characterizing Traffic Load, Characterizing Traffic Behaviour, Characterizing Quality of Service Requirements.
3. **Designing a Network Topology**: Hierarchical Network Design, Redundant Network Design Topologies, Modular Network Design, Designing a Campus Network Design Topology, Virtual LANs, Wireless LANs, Redundancy and Load Sharing in Wired LANs, Server Redundancy, Workstation-to-Router Redundancy, Designing the Enterprise Edge Topology, Secure Network Design Topologies. **Designing Models for Addressing and Numbering:** Guidelines for Assigning Network Layer Addresses, Designing a Model for Naming.
4. **Selecting Switching and Routing Protocols**: Making Decisions as Part of the Top-Down Network Design Process, Selecting Switching Protocols, Selecting Routing Protocols, IP Routing.
5. **Developing Network Management Strategies:** Network Management Design, Network Management Architectures, Selecting Network Management Tools and Protocols. **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.

Pre-requisite Courses: UE17CS301 – Computer Networks.

Reference Book(s):

1. “Top-Down Network Design”, Priscilla Oppenheimer, Cisco Press; 3rd Edition

UE17CS347:
RECONFIGURABLE COMPUTING (4-0-0-0-4)

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

- Understand the motivation for reconfigurable computing.
- Introduction to internal structure of reconfigurable devices.
- Understand design of accelerator logic.

- Learn industry standard development tools.

Course Outcomes:
At the end of the course, the student will be able to:
- Comprehend the rationale behind reconfigurable computing.
- Build computation structures to accelerate processing.
- Use industry standard tools to implement accelerator logic.
- Design and implement accelerator logic to speed up an application.

Course Content:
1. **Introduction:** Business and Technical Motivations (such as Moore's law challenges and big data computation requirements) for Reconfigurable Computing.
2. **FPGA Architecture:** Architecture and Organization (Logic Blocks, Interconnect, Memory and Arithmetic Blocks) of Modern FPGA (Field Programmable Gate Array) Reconfigurable Devices.
3. **Accelerator Logic**: Design of Logic Structures, Memory Structures and State Machines and Their Implementation using Structural Verilog HDL.
4. **Development Software:** Industry Standard Tools for Compiling, Optimizing, Simulating and Debugging Accelerator Logic.
5. **Applications:** Hands on Programming of Basic Computation Structures and At Least One Application That Performs Much Faster Than Software.

Pre-requisite Courses: None.

Reference Book(s):

1. "Digital System Design with FPGA: Implementation Using Verilog and VHDL", 1st Edition, Cem Ünsalan and Bora Tar, McGraw-Hill Education, 2017.

UE16CS401:
OBJECT ORIENTED MODELING AND DESIGN (4-0-0-0-4)

Course Objectives:
The objective(s) of this course is to,
- Introduce students to Object Oriented Programming, Object Oriented Analysis and Modeling using the Unified Modeling Language (UML).
- Familiarize them with the models used in UML, including static as well as dynamic (behavioural) models.
- Make students appreciate the importance of system architecture and system design in product development.
- Introduce students to important design principles and patterns including GRASP and SOLID.

Course Outcomes:
At the end of the course, the student will be able to:
- Use the concepts of classes and objects in Object Oriented Programming. Use UML to model a complex system by defining actors and use cases.
- Construct Class Models and analyze the dynamics of a system using Activity, Sequence, State and Process models
- Depict the architecture of a software system by using component and deployment models and design a database based on a class model.

- Use GRASP and SOLID principles in the design of software.
- Apply software design patterns in a variety of situations.

Course Content:

1. **Introduction, Use Cases and Class Models:** Introduction to Object Oriented Programming, Introduction to Modeling, Introduction to UML, Use Case Models, Application to Case Study
2. **Class Models and Dynamic Models:** Class Modeling, Object Constraint Language, Advanced Class Modeling, Activity Models, Business Process Models, Sequence Models, ATM Case Study: Application Class / Interaction Models, State Models, Advanced State Models, Relationship between Class and State Models, Application to Case Study.
3. **System Design:** Component and System Design, Class Design, Implementation Models, Deployment Models, Object-Relational Mapping, Database Design.
4. **Object Oriented Design Principles:** GRASP (General Responsibility Assignment Software Patterns) and SOLID (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion).
5. **OO Design Patterns:** What Design Patterns Are, How Design Patterns Solve Problems, How to Select a Design Pattern, How to Implement a Design Pattern, Introduction to Widely Used Design Patterns including Creational, Structural, and Behavioural Patterns.

Pre-requisite Courses: None.

Reference Book(s):

1. **"**Object-Oriented Modeling and Design with UML", Michael R Blaha and James R Rumbaugh, 2nd Edition, Pearson.
2. "Applying UML and Patterns", Craig Larman, Pearson, 3rd Edition
3. "The Unified Modeling Language User Guide", Grady Booch, James Rumbaugh and Ivar Jacobso, Pearson, 2nd Edition.

UE16CS402:
SOFTWARE ENGINEERING (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- 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 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 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:

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, etc. **Software Project and Product Management Overview:** Planning a Software Development Project with Overview of Different Aspects of Software Engineering Management and Process Maturities. **Requirements Engineering and Modeling:** Requirements Engineering Tasks, Requirements Documentation / Specification and Management, Requirements Traceability**.**
2. **Software Architecture:** Software Architecture, Software Life Cycle, Architecture Design, Architectural Views, Architectural Styles, the Unified Modeling Language. **Software Design:** Classical Design Methods, Object Oriented Analysis and Design Methods, Design Patterns, Service Orientation - Service Descriptions and Service Communication, Service Oriented Architecture. **Implementation:** Coding Standards and Guidelines, Code Review / Peer Review, Patching and Patch Management.
3. **Change and Build Management:** Elements of a Configuration Management System, Baselines, Repository, SCM Process, Configuration Management Plan, Management of Code Versions, Release Versions, Exposure to Code Management Tools / Build.
4. **Software Testing:** Test Objectives, Testing and the 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.
5. **Software Quality:** Managing Software Quality, A Taxonomy of Quality Attributes, Perspectives on Quality, The Quality System, Software Quality Assurance, The Capability Maturity Model, Personal Software Process. **Other Eng. Topics:** CBSE, Software Metrics, Software Engineering in a Global Environment, Software Estimation, Software Engineering and Hacking, Ethics in Software Engineering. **DevOps:** Introduction, Overview and Component.

Pre-requisite Courses: None.

Reference Book(s):

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

UE16CS403:
WEB TECHNOLOGIES II (4-0-0-0-4)

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

- Provide an understanding of designing advanced Web Applications using AJAX techniques.
- Provide an introduction to challenges involved in improving performance of rich web sites.
- Learn the basics of Web Services and Reverse AJAX techniques.
- Provide an overview of various Web Frameworks.
- Provide a basic overview of non-functional side of the WWW.

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

- Design Advanced Web Applications, with techniques like AJAX, Server-Push, etc.
- Use Node.js as a Server-Side Framework to develop web sites that provide fast and reliable content.
- Develop RESTful Web Services and consume them.
- Develop robust web sites that are immune to malicious web attacks.

Course Content:

1. **AJAX:** JS objects, prototype inheritance, Ajax **-** Hidden Frames Technique, Image-Based AJAX, Dynamic Script Loading, XMLHttpRequest, Cross-Domain Access (CORS), Maintaining History in AJAX Calls, and Fetching Binary Data with XHR.
2. **AJAX Patterns and Data Formats** – Predictive Fetch, Multi-Stage download, Periodic Refresh and Fallback Patterns, Submission Throttling. Introduction to XML, Parsers, Styling RSS / Atom Feeds, JSON and XML, JSON vs XML.
3. **Reverse AJAX:** COMET, HTTP Streaming and Long Poll, iFrames Technique, Using XMLHttpRequest, HTML5 Server Sent Events, Overview of COMET Frameworks, Web Sockets. **Web Services**: Principles of REST, Architecture of SOAP-Based Services, REST vs. SOAP Based Services, Building and Invoking RESTful Web Services using PHP, Introduction to Micro Services.
4. **Non-Functional Aspects of the Web:** Performance Considerations - Timeouts, Retries, Handling Server Errors, Multiple Requests, The HTTP 1.1 Two Connection Limit, Race Conditions, Caching on Client Side, Compression of Data, Request Management, Scalability. HTTP 2.0 – New Features, HTTP 2.0 vs 1.1. Security – Web Attack Surfaces, Reconnaissance Review, Various Vulnerabilities and Precautions, SQL Injection, XSS, XST, CSRF, HTTP Authentication and its Configuration in Apache.
5. **Web Frameworks:** Overview of Popular Frameworks. **Node.js**: Basics, Callbacks, HTTP Responses, File and Database Access. **Angular JS:** Introduction, Directives and Controllers, Forms, Inputs and Services, Server Communication with $http, Filters.

Pre-requisite Courses: UE16CS204 – Web Technologies I.

Reference Book(s):

1. "Professional AJAX", Nicholas C. Zakas et. al, 2nd Edition, Wiley Publishing, 2007.
2. "Angular JS Up and Running", Shyam Sheshadri and Brad Green, O Reily, 1st Edition, 2014.
3. "AJAX: The Complete Reference", Thomas A Powell, McGraw Hill, 2008.
4. Handouts for Web Sockets, HTTP 2.0 protocol, Node.js.

UE16CS411:
ENTERPRISE RESOURCE PLANNING (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- To learn the strategic importance of Enterprise Resource Planning systems in industry
- To learn the basics of ERP, the modules of ERP
- To learn the key selection criteria and implementation issues & risks of ERP
- To be aware of ERP related technologies
- To be aware of 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 criteria to select ERP Package
- Apply criteria to select ERP Consulting Partner
- Systematically develop plans for an ERP Implementation project and
- Identify critical success factors and associated risks.

Course Content:

1. **Introduction to Enterprise Resource Planning Systems:** Introduction, Value Chain Framework, Problems with Disintegrated Data in an Organization, Evolution of ERP Systems, Role of ERP Systems in an Organization, Are ERP Systems Different from Traditional Information Systems?, Scope of ERP Systems, General Model of Business and Role of ERP, Major ERP Players, Implementations in India. **Life Cycle of an ERP Implementation Project:** Introduction, Life Cycle of an ERP Project, ERP Project Teams, Implementation Methodologies, Deployment Methods. **Benefits and Cost of an ERP System:** Introduction, Benefits, Cost of an ERP Implementation, Cost-Benefit Analysis.
2. **Change Management:** Introduction, People Issues: Are You Ready for ERP?, Factors that Influence Pre-implementation Attitude, How to Enhance Attitude?, Change Management Strategies to Handle Organizational Issues, Creating a Change Management Strategy to Handle Organizational Issues, Tools for Assessing the Organizational Changes. **Re-Engineering:** Introduction, Processes and their Characteristics, Life Cycle of a BPR Project, Life Cycle of an IT-driven BPR Project, Re-engineering Examples, Case Studies. **Business Process Modelling and Business Modelling**: BPM Introduction, Business Process Hierarchy, Standards for Business Processes and Modelling, Process Modelling Maturity and Multi-Dimensional Modelling, Process Modelling Software, Business Modelling, Integrated Data Modelling.
3. **ERP Functional Modules: Human Capital Management:** Introduction, Human Capital Management Systems, Leading HR Solutions from ERP Vendors, Strategic Vs. Operational HR Processes and HR Outsourcing, Employee Health and Safety. **Financial Management:** Introduction, ERP Financial Application, Financial Modules in Detail. **Procurement and Inventory Management:** Procurement, Inventory Management. **Production Planning and Execution:** Understanding MRP II Concepts, How ERP PP module supports MRP II Processes, Critical Master Data Elements, Managing different Production Scenarios

4. **ERP Selection:** Introduction, ERP System Selection Team, ERP Solution and Vendor Selection, Information Gathering, Preliminary Filtering, Parameters for ERP Selection, Prepare and Release Request for Proposal (RFP), Gap Analysis, AHP for ERP Selection. **Managing an ERP Project:** Introduction, Success of an ERP System is Multi-dimensional, Critical Success Factors, Risk Associated with an ERP Project, Measuring Performance of ERP System (using Balanced Scorecard).
5. **ERP and Related Technologies:** Introduction, ERP on Cloud, Supply Chain Management (SCM), Customer Relationship Management (CRM), Product Life Cycle Management (PLM), Data Warehousing, Business Intelligence (DW-BI). **Introduction to Commercial ERP Software**: Introduction, Indian Market, SAP, Oracle, PeopleSoft, JD Edwards, MS Dynamics. Implementations in India; Articles and Cases. Introduction to ERP for SME's.

Pre-requisite Courses: None.

Reference Book(s):

1. "Enterprise Resource Planning: A Managerial Perspective", Veena Bansal, Pearson Education India, 2013.
2. "Enterprise Resource Planning- Text & Cases", Rajesh Ray, Tata McGraw Hill, New Delhi, 2011.
3. "ERP Demystified", Alexis Leon, McGraw Hill Education, 3rd Edition, 2014.

UE16CS412:
ALGORITHMS FOR INFORMATION RETRIEVAL (4-0-0-0-4)

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

- Understand the architecture, models and algorithms used in Information Retrieval.
- Understand the basic principles and implementation of Indexing and Search.
- Understand how web search works.
- Understand the use of machine learning in Information Retrieval.
- Become familiar with applications and latest trends in Information Retrieval.

Course Outcomes:
At the end of the 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.
- Implement components of a web search system.
- Apply Machine Learning techniques in Information Retrieval Systems.
- Describe modern applications and research trends in Information Retrieval.

Course Content:

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.
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.
3. **Evaluation of IR / Other IR Models:** Performance Measurement, Relevance

Feedback, Query Expansion, Other IR Models.

4. **Web Search:** Web Search Basics, Economic Model of Web Search, Search User Experience, Web Crawling and Indices, Link Analysis, The PageRank Algorithm, Building a Complete Search System.
5. **Applications of IR:** Text Classification and Clustering, Snippet Generation, Summarization, Topic Detection and Tracking, Question Answering, Personalization.

Pre-requisite Courses: UE16CS251 – Design and Analysis of Algorithms.

Reference Book(s):

1. "Introduction to Information Retrieval", Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, ISBN: 9781107666399, Cambridge University Press, 2009.
2. "Modern Information Retrieval: The Concepts and Technology behind Search", Ricardo Baeza -Yates and Berthier Ribeiro – Neto, ACM Press Books, 2nd Edition, 2011.
3. "Search Engines: Information Retrieval in Practice", Bruce Croft, Donald Metzler and Trevor Strohman, Addison Wesley, 1st Edition, 2009.
4. "Information Retrieval: Implementing and Evaluating Search Engines", Stephen Buettcher, Charles L A Clarke and Gordon V Carmack, MIT Press, 2010.

UE16CS413:
COMPUTER VISION (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- To create an awareness of the imaging fundamentals in terms of acquisition, storage and display.
- To gain an insight into the mathematical transforms necessary for binary and gray scale images.
- To study the quality of the image and the enhancement of images in spatial and frequency domains.
- To design and evaluate the methods of mathematical morphology and image segmentation.
- To translate the techniques of gray scale images to 3D/color images.

Course Outcomes:

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

- Identify typical defects in an image and assess a suitable technique for processing the image to mitigate/ remove the defect and enhance the image.
- Design methods to automatically extract regions of interest from each other and the background either in binary, grayscale or color images.
- Extract the gist of any method in literature for processing of images for a particular application and be able to explain this clearly, extending their learning of concepts such as histogram equalization to local/ adaptive histogram equalization and enhancement techniques in the spatial and frequency domains to block processing of images - realized through seminars in class.

- Assess hardware available for acquisition and viewing of images for advantages and limitations and be able to identify a suitable imaging modality for a given application.
- Implement basic techniques using open source tools and take up a problem, conceptualize the solutions and implement in a programming language of their choice - realized through a hands-on workshop, assignment and class project.
- Application of spatial and frequency domain processing for computer vision.

Course Content:

1. **Introduction to Computer Vision:** A Brief History of Computer Vision, The Digital Camera, Point Operators, Linear Filtering, Neighborhood Operators, Basics of Frequency Domain Processing.
2. **Feature Detection and Matching and Segmentation:** Features - Feature Extraction, Binary Object Features, Histogram Features, Color Features, Spectral Image Features, CornerDetection, Scale Invariance, Projections, Segmentation, Points, Lines, Split and Merge, Mean – Shift.
3. **Computational Photography:** Photometric Calibration, High Dynamic Range Imaging, Super Resolution and Blur Removal, Basics of Image Matting and Texture Analysis.
4. **Stereo and 3D Reconstruction**: Stereo - An Introduction, Epipolar Geometry, Sparse and Dense Correspondence, Local Methods, Feature Tracking and Optical Flow.
5. **Image Indexing and Recognition: Image indexing, image search and retrieval,** Object Detection, Face Recognition, Category Recognition, Context and Scene Understanding, Video Summarization.

Pre-requisite Courses: None.

Reference Book(s):

1. “Computer Vision: Algorithms and Applications”, Richard Szeliski, Springer, 2nd Edition, 2010.
2. “Computer Vision – A Modern Approach”, Forsythe and Ponce, Pearson, 2nd Edition, 2011.
3. “Dictionary of Computer Vision and Image Processing”, R. B. Fisher, T. P. Breckon, K. Dawson-Howe, A. Fitzgibbon, C. Robertson, E. Trucco, C. K. I. Williams. Chichester, West Sussex: John Wiley & Sons Inc., 2014.

UE16CS414:
TOPICS IN DEEP LEARNING (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- 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 the course, the student will be able to:

- Implement Machine Learning techniques with TensorFlow and Keras.
- 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.
- Develop simple game engines using Reinforcement Learning.

Course Content:

6. **TensorFlow and Keras and Reinforcement Learning**: Brief overview of Deep Learning 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.
7. **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
8. **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 Naïve Bayes, Unsupervised Feature Learning – Autoencoders and Variations.
9. **CNN, GAN and Transfer Learning: CNN -** Architecture of CNNs, Filters, Feature Maps, Max-Pool Layers, Other Pooling Types, Case Study: Image Recognition Using CNN – Hands-On Implementation Using Keras. **GAN** - Architecture and Training Methods, Image-Generation, Hands-On Implementation Using Keras. **Transfer Learning** - Motivation, Variations, Use in CNNs.
10. **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.

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

Pre-requisite Courses: UE16CS353 – Machine Learning.

Reference Book(s):

5. "Hands-on Machine Learning with Scikit-Learn and TensorFlow", Aurelian Geron, O'REILLY, 1st Edition, 2017.
6. "Deep Learning with Keras", Antonio Gulli and Sujit Pal, Packt Publishing, 1st Edition, 2017.
7. "Pattern Recognition and Machine Learning", Christopher Bishop, Springer, 1st Edition, 2011 (Reprint).
8. Handouts for SVM, Transfer Learning.

UE16CS415:

WIRELESS NETWORK COMMUNICATIONS (4-0-0-0-4)

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

- To provide an overview of Wireless Communication networks area and its applications in communication engineering.
- To appreciate the contribution of Wireless Communication networks to overall technological growth.
- To understand the various terminology, principles, devices, schemes, concepts, algorithms and different methodologies used in Wireless Communication Networks.

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

- Understand the cellular system design and technical challenges.
- Analyze the Mobile radio propagation, fading, diversity concepts and the channel modelling.
- Analyze the design parameters, link design, smart antenna, beam forming and MIMO systems.
- Analyze Multiuser Systems, CDMA, WCDMA network planning and OFDM Concepts.
- Summarize the principles and applications of wireless systems and standards

Course Content:

1. **Overview of Wireless communication:** Cellular Phone Standards, Cellular Evaluation, Wireless Local Loop (WLL) and LMDS, Wireless Local Area Networks (WLANs), Bluetooth and Personal Area Networks (PANs), Overview of WLAN standards (802.1g/n/ac/ad) and channel management. Handover in WLAN network.
2. **Overview of Wireless Communication-II:** Wifi, Wimax (IEEE 806.16a), IoT Wireless -Topologies, Zigbee Wireless Networks and Transceivers, NFC, 6LoWPAN, Tradeoff between Battery, Bandwidth and Distance. **Wireless Channel Models:** Path Loss and Shadowing Models, Millimeter Wave Propagation, Statistical Fading Models, Narrowband Fading, Wideband Fading Models.
3. Design of WLAN in public place e.g. Cafes, hotels, Railway Stations, Airports. **Impact of Fading and ISI on Wireless Performance:** Capacity of Wireless Channels, Digital Modulation and its Performance. **Overview of SOHO Internet Technologies:** DSL, PON, Cable and Satellite.
4. Adaptive Modulation, Multiple Input/ Output Systems (MIMO).
5. **ISI Countermeasures:** Multicarrier Systems and OFDM, Multiuser and Cellular Systems.

Pre-requisite Courses: UE16CS301 – Computer Networks.

Reference Book(s):

1. “Wireless Communication”, Andrea Goldsmith, First Edition, Cambridge University Press.
2. “Fundamentals of Wireless Communication”, David Tse, Pramod Viswanath, Cambridge University Press, 1st Edition.

3. “Advanced Wireless Communication and Internet: Future Evolving Technologies”, Savo G Glisic, Wiley, 3rd Edition.

UE16CS421:
INFORMATION SECURITY (4-0-0-0-4)

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

- Learn the challenges and pitfalls of Software Development and Secure Programming, across the Web, Mobile Devices and IoT.
- Learn the possible attacks and available remedies.
- Learn about security design and testing best practices.

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

- Understand the security limitations of commonly used Operating Systems, Browsers and Mobile Operating Systems.
- Understand the security limitations of popular programming languages.
- Understand the common security pitfalls in various application development approaches, platforms and how to avoid them.
- Learn how to use some common security testing strategies and Penetration Testing.

Course Content:

1. **Introduction:** Software Threats and Vulnerabilities, OWASP Top 10, SANS Top 25, CVE, etc. Various Type of Attacks like Brute-Force Attacks, DDOS, Phishing, Credentials Misuse, Malware, etc. Significance/ Importance of Secure Coding, Secure Coding Terminology, Secure Coding Principles, Threat Modeling, Secure Coding Resources (Both online and offline).
2. **Secure Software Development and Programming:** Common Operating Systems and Their Security Limitations, Common Programming Languages and Their Security Limitations, Secure Application Development, Database Security, Ransomware, Virus, Malicious Code, Testing and Prevention, Best Practices.

   2.1. C++ Catastrophe, Calls to Delete, Constructors, Lack of Reinitialization, Ignorance of STL, Pointer Initialization, Testing Techniques and Defensive Measures. Catching Exceptions - Overview, Structured Exception Handling, Signal Handling, Failure to Handle Errors Correctly, Yielding Too Much Information, Ignoring Errors, Misinterpreting Errors, Using Useless Return Values, Using Non-Error Return Values. Platform Security - Code Integrity and Code Signing, Secure Boot, Measured Boot and Root of Trust, Security threats from peripherals, e.g., DMA, IOMMU.

   2.2. Vulnerabilities and Exploits like Buffer Overflows, SQL Injection, Insecure Direct Object References, Security Misconfiguration, Sensitive Data Exposure

3. **Web Application Security Issues:** Challenges, Browser Security, SQL Injection, Cross-Site Scripting, Cross-Site Request Forgery, Session Hijacking, TLS Stripping, Cross-Site Scripting (XSS), Broken Authentication and Session Management.

3.1. HTTP Security: Overview of HTTP Security, MITM Attacks and Solutions, HTTP Security Headers: CSP (Content-Security-Policy), HSTS (HTTP Strick Transport Security), HPKP (HTTP Protocol Key Pining), X-Frame-Options, X-XSS-Protection, X-Content-Type-options, CORS (Cross Origin Resource Sharing), HTTP/ 2 and Security Challenges, HTTP Security Considerations - Transfer of Sensitive Information and Its Encoding, Privacy Issues and HTTP Authentication.

3.2. Executing Code with Too Much Privilege, Examples and Defensive Measures, Failure to Protect Stored Data, Weak Access Controls on Stored Data, Weak Encryption of Stored Data, Use of Weak Password-Based Systems, Password Compromise Password, Change Policies, Password Failure Error Display Policies, Retrieval of Forgotten Passwords, Default Passwords and Replay Attacks, Storing Passwords and Alternatives, Password Verifiers Zero Knowledge, Brute-Force Attacks Against Password Verifiers.

4. **Mobile Application Security** - Android and iOS Security, App Security, Secure Boot, Data Exfiltration, Cloud and IoT Application Security.
5. **Countermeasures - Tools, Frameworks, and Services:** Secure Coding Standards, Secure Coding Best Practices/ Patterns, Intercepting Validators, Sanitization, Session Management, Authentication, Encryption, Password Management, Access Control, Error Handling and Logging, File Management, Memory Management, Microsoft Secure Development Process (SDP), Static Analysis Tools, Dynamic Analysis Tools, Web Application Security Frameworks, Java-Based Enterprise Application Security Frameworks, Outsourcing, Vulnerability Tracking.

Pre-requisite Courses: UE16CS331 – Computer Network Security.

Reference Book(s):
1. Computer Security – Principles and Practice", William Stallings, 3rd Edition.

UE16CS422:
WEB SERVICES (4-0-0-0-4)

Course Objectives:
The objective(s) of this course is to,
- To introduce Service Oriented Architecture.
- To provide a comprehensive introduction to Web Services.
- To impart techniques on building RESTful and SOAP-based Services.
- To educate the student on the security aspects of Web Services.

Course Outcomes:
At the end of the course, the student will be able to:
- Build RESTful Web Services
- Build SOAP based Web Services
- Design RESTful APIs
- Demonstrate basic security mechanisms in Web Services.
- Invoke RESTful and SOAP-based Web Services.

Course Content:

1. **Introduction:** Web Services and SOA, Overview of HTTP, REST.
2. **RESTful Web Services - The Service Side:** RESTful Service as an HTTP Servlet, as a JAX-RS Resource, Generating XML and JSON Responses, Restlet Resources, WebService Provider.
3. **RESTful Web Services - The Client Side:** Clients against Amazon E-Commerce Service, RESTful Clients and WADL Documents, JAX-RS Client API, JSON for Javascript Clients, JSONP and Web Services, jQuery, AJAX Polling.
4. **SOAP-Based Web Services:** SOAP-Based Services, Java Clients, WSDL Service Contract, SOAP-Based Clients against Amazon E-Commerce Services, Asynchronous Clients.
5. **Web Services Security:** Wire-Level Security - Service and Client-Side, HTTPS – Encryption, Decryption, Handshake, etc. in Detail, Container-Managed Security, WS-Security.

Pre-requisite Courses: UE16CS204 – Web Technologies I.

Reference Book(s):

1. "Web Services Up and Running", Martin Kalin, 2nd Edition, O Reilly, 2013.

UE16CS423:
ALGORITHMS FOR INTELLIGENT WEB (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- To get an understanding of search algorithms.
- To learn recommendation algorithms and their working.
- To get an insight into clustering algorithms.
- To introduce classification algorithms and their use in today's applications.
- To get an idea of different web applications.

Course Outcomes:

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

- Analyze different search algorithms and make comparison
- Suggest different recommendation algorithms
- Implement various clustering algorithms
- Analyse different intelligent applications
- Effectively use classification algorithms

Course Content:

1. **Introduction and Search Algorithms:** Introduction to Web and Intelligent Web Applications, Examples of Intelligent Web Applications, Intro to AI and Machine Learning, Intro to Search and Information Retrieval, Lucene as a Search Engine, Improving Search Results, Link Analysis and PageRank, Other Search Algorithms, Scalability Issues in Search.
2. **Recommendation Algorithms:** Distance and Similarity, Distance and Similarity, Recommendation algorithms, Types of Recommendations, Workings of Sample Systems, Workings of Sample Systems, Data Normalization and Correlation Coefficients.
3. **Clustering Algorithms:** Introduction to Clustering, Grouping in SQL, Clustering Algorithms, Types of Clustering Algorithms, Example Algorithms, Applications of Clustering, Scalability Issues in Clustering.

4. **Classification Algorithms:** Classification Theory, Category Lists, Taxonomy, Folksonomy and Ontology, Classification by Tagging, Automatic Classification and Routing, Types of Classification Algorithms, Hybrid Classifiers, Sample Applications, Practical Issues in Classification.
5. **Intelligent Web Applications:** 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.

Pre-requisite Courses: UE16CS251 – Design and Analysis of Algorithms.

Reference Book(s):

1. "Algorithms of the Intelligent Web", Haralambos Marmanis, Dmitry Babenko, Manning Publishers, 2011.

UE16CS424:
SOCIAL NETWORK ANALYTICS (4-0-0-0-4)

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

- Provide students background on 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 variety of real world scenarios by modelling them as games**.**

Course Content:

1. **Background and Fundamentals of network analysis:** Introduction to Networks and Examples, Ego-centric Networks, Exchange Networks, Graph-Theory, Directions and Weights, Adjacency Matrices, homophily, Tie-strengths and structural holes**.** Representing and Measuring Networks: Degree distribution, diameters, path-lengths, centrality, closures, clustering
2. **Models of Network formation:** Random Networks, Small World, Growing Random Networks, Growth Models, Distribution of expected degrees, Preferential attachment, Fat tails, Power Laws, Fat Tails, Scale-free networks, Affiliation

Networks, Cliques and Cores, Cohesion, Communities and Community Detection Algorithms

3. **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, Assortativity, Percolation and Robustness of Networks, Effects of communities and centralities on diffusion
4. **Strategic Networks:** Economic Game Theoretic Models of Network Formation, Connections Model, Pair-wise Stability, Efficient and Pareto-efficient networks, Externalities and Co-author Models, Pair-wise Nash Stability, Complements and Substitutes.
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**.**

Pre-requisite Courses: UE16MA251 -Linear Algebra, UE16CS202 – Data Structures

Reference Book(s):

1. "Introduction to Social Network Methods", Robert A Hanneman, University of California Riverside, 2005.
2. "Social and Economic Networks", Mathew O Jackson, Princeton University Press, 2008.
3. "Networks, Crowds, and Markets: Reasoning About a Highly Connected World", D Easley and J Kleinberg, Cambridge University Press, 2010.
4. Networks – An introduction MEJ Neumann, Oxford University Press 2010.
5. "Social Network Analysis for startup", Maksim Tseovat and Alexander Kouzntsov, O'Reily, 2015.

UE16CS425:
COMPUTER SYSTEMS PERFORMANCE ANALYSIS (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Provide students background on the art of system performance evaluation
- Train students to observe workload, workloads 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

Course Content:

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
2. **Workloads, Workload Selection and Characterization:** Types of Workloads, Addition Instructions, Instruction Mixes, Kernels, Synthetic Programs, Application Benchmarks, Popular Benchmarks, Workload 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.
3. **Experimental Design and Analysis, Simulation:** Introduction - Terminology, Common Mistakes in Experiments, Types of Experimental Designs, 2^k Factorial Designs, Concepts, Computation of Effects, Sign Table Method for Computing Effects, Allocation of Variance, General 2^k Factorial Designs, $2^k r$ 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.
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 and Bottleneck Analysis.
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.

Pre-requisite Courses: UE16CS253 – Microprocessor and Computer Architecture

Text book:

1. "The Art of Computer Systems Performance Analysis", Raj Jain, John Wiley and Sons, 2007.

UE16CS426:
DESIGN PATTERNS (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Impart design principles beyond coding.
- Inculcate good habits in design.
- Make the participants appreciate what to do and what not to do.
- Compare alternate design solutions.
- Appreciate the intricacies of design.

Course Outcomes:

At the end of the 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.
- Select where idioms can be applied as opposed to design patterns.
- Identify where not to apply design patterns.
- Reliably refactor a large piece of software.

Course Content:

1. **Design Principle:** Interface and Implementation, Open Closed Principle, Liskov Substitution Principle, Dependency Inversion Principle, Integration Segregation Principle. **Architectural and Package Principles.**
2. **Idioms:** Handle Body Idiom (PIMPL), Reference Counting, Named Constructor Idiom, Telescoping Constructor, Bean Pattern for Construction, Destruction Idiom.
3. **Design Patterns:** GOF Patterns, Constructional Patterns, Structural Patterns, Behavioural Patterns.
4. **Patterns:** Beyond GOF, Persistence, Multi-threading.
5. **Introduction to Anti-patterns and Refactoring.**

Pre-requisite Courses: UE16CS251 – Design and Analysis of Algorithms.

Reference Book(s):

1. "Design Principles and Design Patterns", Robert C Martin, 2000.
2. "Design Patterns: Elements of Reusable Object-Oriented Software", Gamma et al, Addison Wesley, 1994
3. "Java Design Patterns", James W Cooper, Addison Wesley, 2000.
4. "AntiPatterns - The Survival Guide to Software Development Processes", Alexander Shvets, Online Reference at http://bit.ly/2e4nxzd).

UE16CS427:
AUTONOMOUS MOBILE ROBOTICS (4-0-0-0-4)

Course Objectives:

The objective(s) of this course is to,

- Enable the students to have hands-on experience with mobile robots
- Understand the physics of motion
- Write program to operate mobile robots
- Use mobile robots for accomplishing high level tasks
- Enrich the mobile robotics applications to serve the society

Course Outcomes:

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

- Operate mobile robots autonomously or/and semi-autonomously
- Write programs to guide the paths of mobile robots
- Perform assigned tasks using mobile robots
- Apply the fundaments of mobile robotics for their projects
- Innovate workable ideas to address the challenging problems in the society.

Course Content:

1. **Introduction and Locomotion:** Introduction to mobile robots (MR), Applications, Key issues in locomotion, Legged mobile robots, Wheeled mobile robots

2. **Kinematics:** MR kinematic models, MR constraints, MR maneuverability, MR workspace, MR control basics
3. **Perception:** Sensor for MR, Representing uncertainty, Error propagation, Feature extraction – range data, Feature extraction – vision
4. **Localization:** Challenges in localization, Map representation, Map-based localization, Other localization methods, Automatic map building
5. **Planning and Navigation:** Path planning, Obstacle avoidance, Navigation architectures, Control localization, Techniques for localizations

Pre-requisite Courses: None.

Reference Book(s):

1. "Introduction to Autonomous Mobile Robots", Roland Siegwart, Illah Nourbakhsh, The MIT press 2004 (eBook is freely available).
2. "Mobile Robotics: Mathematics, Models, and Methods", Alonzo Kelly, Cambridge University Press, 2013.
3. "Robot Programming: A Practical Guide to Behavior-Based Robotics", Joe Jones, Daniel Roth, McGraw-Hill Education TAB; 1st Edition, 2004.

UE16CS451:
INTRODUCTION TO SOFTWARE TESTING (2-0-0-0-2)

Course Objectives:

The objective(s) of this course is to,

- Introduce the concepts of Software Quality and types of testing
- Familiarize with different levels of testing – Unit, Integration, System and Acceptance Testing
- Familiarize with Non-functional Testing and Regression Testing
- Introduce to software testing tools
- Understand the 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

Course Content:

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 SDLC and other new methods of Software development like Agile, SCRUM etc. Modified V Model for testing requirements in a project. SQA processes, tools and techniques for Test Life Cycle. Software Quality Measurements. Classification of testing types based on method / Requirement / target / needs.

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.
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 Box Testing:** Definition and overview.
4. **Acceptance, Non-functional and Regression Testing:** Acceptance Testing Overview, Acceptance Testing Approaches – Design & Architecture based, Business vertical / customized instances, Deployment testing, Types: User Acceptance Testing, Alpha and Beta Testing.
   Non Functional Testing: NFT Overview, Scalability, Reliability and Stress testing. Performance Testing Overview, methodology for performance testing. **Regression Testing:** Definition, Types of regression testing, when and how to do regression testing. Introduce testing tools and defect tracking tools.
5. **Testing Tools:** JUnit, JMeter, Monkey Talk, Appium, Robotium, Selenium, Selendroid, UI Automator and Magneto. Discuss the tools and do a comparative study. Hands on in a 2 hour lab session

Pre-requisite Courses: None.

Reference Books:

1. "Software Testing – Principles and Practices" , Srinivasan Desikan and Gopalaswamy Ramesh, Pearson, 2006
2. " Foundations of Software Testing ", Aditya Mathur, Pearson, 2008
3. "Software Testing, A Craftsman's Approach ", Paul C. Jorgensen, Auerbach, 2008

UE16CS452:
INTRODUCTION TO BUSINESS (2-0-0-0-2)

Course Objectives

- Understand the different types of business ownership
- Understand how a modern business organization works
- Understand the importance of marketing and the activities involved in it
- Understand how a business organization is run and managed
- Understand the activities required to set up a start-up company

Course Outcomes

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

- Describe the various legal forms of business ownership
- Explain the functions and working of a business organization
- Describe the processes involved in product/services strategy
- Explain the role of management in a business organization
- Explain the factors that can make a startup company successful

Course Content

1. **Entrepreneurship and Start-ups: Entrepreneurship** - Meaning, Evolution and development. Functions of an entrepreneur. **Start-Ups:** Evolution of start-ups, Important factors in startups: Idea, Team, Business Model, Funding Options, Timing. Startup Culture, Startup ecosystems, Startup environment in India, Case Studies of successes and failures.
2. **Introduction and Legal forms of Business:** Historical context of how businesses evolved into their present forms. Family Businesses; Forms of ownership (proprietorship, partnerships, and corporations).
3. **Business Functions:** Typical functions of any business organization: Production, Finance, Accounting, Human Resource Management, Research and Development, and Marketing. Importance and Management aspects of all functions. Business reporting - Balance Sheets and Profit/Loss accounts and modern methods of business reporting.
4. **Marketing:** The activities for creating, communicating, delivering, and exchanging offerings that have value for customers, clients, partners, and society at large. Marketing is used to create the customer, to keep the customer and to satisfy the customer. How to tailor a product or service to a target market. Digital marketing and its effect on the traditional marketing methods.
5. **Management:** Setting the strategy of an organization and coordinating the efforts of its employees to accomplish its objectives through the application of available resources, such as financial, natural, technological and human resources.

Pre-requisite Courses: None

Reference Books:

1. Introduction to Business, Student Edition (McGraw Hill) https://www.amazon.com/Introduction-Business-Student-BROWN-BUSINESS/dp/0078747686
2. Entrepreneurship Simplified (From Idea to IPO) by Ashok Soota and SR Gopalan, Penguin Random House India, 2016, https://www.flipkart.com/entrepreneurship-simplified-idea-ipo/p/itmemhky2qrg7zqn?pid=9780670088959
3. The single biggest reason why startups succeed | Bill Gross (TED Talk) https://www.youtube.com/watch?v=bNpx7gpSqbY

UE16CS453:
RESEARCH METHODOLOGY (2-0-0-0-2)

Course Objectives:

The objective(s) of this course is to,

- Define research and identify the systematic steps to be followed.
- Identify the overall process of designing a research study from inception to its report.
- Impart familiarity with ethical issues in educational research, including those issues that arise in using 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.
- Select and define appropriate research problem and parameters.
- Prepare a project proposal (to undertake a project).

Course Content:

1. **An Introduction:** Meaning, Objectives and Characteristics of research - Research methods Vs Methodology -Types of research - Descriptive Vs. Analytical, Applied Vs. Fundamental, Quantitative Vs. Qualitative, Conceptual Vs. Empirical - Research process - Criteria of good research .
2. **Research Design:** Research design and methods – Research design – Basic Principles- Need of research design ––Features of good design – Important concepts relating to research design.
3. **Sampling design** - Steps in sampling design - Characteristics of a good sample design - Types of sample designs - **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.
4. **Testing of hypotheses** - Basic concepts - Procedure for hypotheses testing, flow diagram for hypotheses testing, Important parametric test - Chi-square test .
5. **Interpretation and report writing** - Techniques of interpretation - Structure and components of scientific reports - Different steps in the preparation - Layout, structure and language of the report - Illustrations and tables - Types of report - Technical reports and thesis

Pre-requisite Courses: None.

Reference Book(s):

1. “Research Methodology: Methods and Techniques”, C R Kothari, Gourav Garg, New Age International Publishers, 3rd Edition, 2014.
2. “Research Methods for Engineers”, David V Thiel, Cambridge University Press, 2014.

UE16CS454:
ADVANCED MACHINE LEARNING (2-0-0-0-2)

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

- Introduce learning theory of Machine Learning
- Familiarize with complexity theory of techniques in Machine Learning practice
- Understand the notions of computational learning and convex learning
- Become conversant with stochastic optimization technique and their applicability

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

- Derive and implement optimization algorithms for the advanced models.
- Have a good understanding of the two numerical approaches to learning (optimization and integration) and how they relate to the models described in the course.

- Have an understanding of how to choose a model to describe a particular type of data.
- Know how to evaluate a learned model in practice.
- Understand the mathematics necessary for constructing novel machine learning solutions.
- Appreciate complexity and computational learning theory of sophisticated models

Course Content:

1. **Introduction & Basics** Gentle Introduction to ML- What Is Learning? When Do We Need Machine Learning? Types of Learning. Relations to Other Fields A Formal Model: The Statistical Learning Framework. Empirical Risk Minimization. Something May Go Wrong Overfitting. Empirical Risk Minimization with Inductive Bias
2. **Learning theory:** Formal Learning: PAC Learning, A More General Learning Model, Releasing the Realizability Assumption: Agnostic PAC Learning, The Scope of Learning Problems Modeled, Learning via Uniform Convergence: Uniform Convergence Is Sufficient for Learnability, Finite Classes Are Agnostic PAC Learnable, Bias-Complexity Tradeoff: The No-Free-Lunch Theorem, No-Free-Lunch and Prior Knowledge, Error Decomposition, VC Dimensions: Infinite-Size Classes Can Be Learnable, The VC-Dimension, Examples, Threshold Functions, Intervals, Axis Aligned Rectangles, Finite Classes, VC-Dimension and the Number of Parameters, The Fundamental Theorem of PAC learning, Proof of Theorem, Sauer's Lemma and the Growth Function, Uniform Convergence for Classes of Small Effective Size.
3. **Theory to Algorithms:** Model Selection and Validation: Model Selection Using SRM, Validation, Hold Out Set, Validation for Model Selection, The Model-Selection Curve , k-Fold Cross Validation , Train-Validation-Test Split , What to Do If Learning Fails, Convex Learning problems: Convexity, Lipschitzness, and Smoothness, Convexity, Lipschitzness , Smoothness , Convex Learning Problems, Learnability of Convex Learning Problems, Convex-Lipschitz/Smooth-Bounded Learning Problems, Surrogate Loss Functions, Regularization and Stability: Regularized Loss Minimization, Ridge Regression, Stable Rules Do Not Overfit, Tikhonov Regularization as a Stabilizer, Lipschitz Loss, Smooth and Nonnegative Loss, Controlling the Fitting-Stability Trade-off.
4. **Stochastic Learning:** Gradient Descent, Analysis of GD for Convex-Lipschitz Functions, Subgradients, Calculating Subgradients, Subgradients of Lipschitz Functions, Subgradient Descent, Stochastic Gradient Descent (SGD), Analysis of SGD for Convex-Lipschitz-Bounded Functions, Variants, Adding a Projection Step, Variable Step Size , Other Averaging Techniques, Strongly Convex Functions, Learning with SGD, SGD for Risk Minimization, Analyzing SGD for Convex-Smooth Learning Problems, SGD for Regularized Loss Minimization, Soft-SVM and Norm Regularization, The Sample Complexity of Soft-SVM, Margin and Norm-Based Bounds versus Dimension, The Ramp Loss, Optimality Conditions and \Support Vectors, Duality, Implementing Soft-SVM Using SGD
5. **Spectral Clustering and complexities**: Spectral Clustering, Graph Cut , Graph Laplacian and Relaxed Graph Cuts, Unnormalized Spectral Clustering, Information Bottleneck, A High Level View of Clustering, The Rademacher Complexity , Rademacher Calculus , Rademacher Complexity of Linear Classes, Generalization Bounds for SVM.

Pre-requisite Courses: UE16CS203 - Introduction to Data Science, UE16MA251 – Linear Algebra, UE16CS353 – Machine Learning.

Reference Book(s):

1. “**Understanding Machine Learning-**Shai Shalev-Shwartz and Shai Ben-David”, Indian Edition, CUP, 2017(Reprint).
2. “**Machine Learning: A Probabilistic Perspective- Kevin P. Murphy.** ”, MIT Press 2012
3. Appropriate handouts and additional material to be provided in course pack.