Syllabus of B.E. Sem. III (Computer Science & Engineering)

3KS01/3IT01/3KE01 ENGINEERING MATHEMATICS-III

Course Objectives:-

1. Find general solutions of linear differential equations with constant coefficients using the roots of the auxiliary equation.
2. Calculate the Laplace Transform of basic functions using the definition.
3. Apply Laplace transform to find solution of linear differential equations. And solve problems related to Fourier Transform
4. Compute and interpret the correlation coefficient.
5. Compute the Analytic function and Complex Analysis.
6. Perform vector differentiation and integration to analyze the vector fields and apply to compute line, surface and volume integrals.

Course Outcomes:

After successfully completing the course, the students will be able to:

1. Demonstrate the knowledge of differential equations and linear differential equations .
2. Apply Laplace transform to solve differential equations.
3. Demonstrate the use of Fourier Transform to connect the time domain and frequency domain.
4. Demonstrate the basic concepts of probability and statistics.
5. Apply the knowledge of Complex Analysis.
6. Apply the knowledge of vector calculus to solve physical problems.

SECTION-A

UNIT-I: Ordinary differential equations:- Complete solution, Operator D, Rules for finding complementary function, the inverse operator, Rules for finding the particular integral, Method of variation of parameters, Cauchy's and Legendre's linear differential equations. (7)

UNIT-II: Laplace Transform:- Definition, standard forms, properties of Laplace transform, inverse Laplace transform, Initial and final value theorem, Convolution theorem, Laplace transform of impulse function, Unit step function, Laplace transforms of periodic function . (7)

UNIT-III: a) Applications of Laplace Transform:- Solution of Linear differential equations, Simultaneous differential equation by Laplace transform method

b) Fourier Transform:- Definition, standard forms, Fourier transforms, properties of Fourier transforms, Convolution theorem, Fourier sine and Fourier cosine transforms and integrals, inverse Fourier transforms.(7)

SECTION-B

UNIT-IV: a) Partial differential equation of first order of following form:- (i)f (p,q) = 0; (ii) f(p,q,z) = 0;

(iii) f (x, p) = g (y,q); (iv) P p + Qq = R (Lagranges Form); (v) z = px + qy + f(p,q) (Clairauts form)

b) Statistics Curve fitting: Least Square Method, Coefficient of Correlations, Lines of Regression. (7)

UNIT-V: Complex Analysis: - Functions of complex variables, Analytic function, Cauchy- conditions, Harmonic function, Harmonic conjugate functions, Milne's Method, conformal mappings (translation, rotation, magnification and bilinear transformation), Expansion of function in Taylor's and Laurent's series. (7)

UNIT–VI: Vector calculus:- Scalar and vector point functions, Differentiation of vectors, Curves in space, Gradient of a scalar point function, Directional derivatives, Divergence and curl of a vector point function and their physical meaning, expansion Formulae (without proof), line, surface, volume integrals, irrotational Solenoidal Vector fields. (7)

TEXT BOOKS:

1. Elements of Applied Mathematics Vol. II by P. N. Wartikar and J.N. Wartikar,
2. Higher Engg. Mathematics by B.S. Grewal.

REFERENCE BOOKS:

1. Advancing Engg. Mathematics by E.K.Kreyzig.
2. A text book of Differential Calculus by Gorakh Prasad.
3. A Text Book of Applied Mathematics by P.N.Wartikar and J.N.Wartikar.
4. Engineering Mathematics by Ravish R Singh, Mukul Bhatt.

3KS02 DISCRETE STRUCTURE AND GRAPH THEORY

Course Prerequisite: Basic knowledge of Mathematics

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Discrete Structure by being able to do each of the following:

1. Use mathematically correct terminology and notation.
2. Construct correct direct and indirect proofs.
3.. Apply logical reasoning to solve a variety of problem

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to

1. Analyze and express logic sentence in terms of predicates, quantifiers, and logical connectives.
2. Derive the solution for a given problem using deductive logic and prove the solution based on logical inference.
3. Classify algebraic structure for a given mathematical problem.
4. Perform combinatorial analysis to solve counting problems.
5. Perform operation on trees data structures.
6. Develop the given problem as graph networks and solve with techniques of graph theory

Unit I: The Foundations: Logic and Proofs **Hours: 8**

Propositions, Truth Tables, Compound Propositions, Logical Operators, Logic and Bit Operations; Logical Equivalences, De Morgan’s Laws, Satisfiability: Applications and Solving Problems; Predicates, Quantifiers: Restricted Domains, Precedence, Logical Equivalences; Rules of Inference for Propositional Logic, Use to Build Arguments.

Unit II: Sets, Functions and Relation **Hours: 8**

Introduction, Venn Diagrams, Subsets, Size of a Set, Power Sets, Cartesian Products, Set Notation with Quantifiers, Truth Sets and Quantifiers, Set Operations ,Functions, Inverse Functions, Compositions and Graphs of Functions, Partial Functions; Sequences, Summations; Countable Sets, An Uncountable Set; Functions as Relations, Relations on a Set, Properties of Relations, Combining Relations; Representing Relations Using Matrices; Representing Relations, Closures of Relations, Equivalence Relations.

Unit III: Algebraic Structures **Hours: 8**

Algebraic Systems: Examples and General Properties; Semigroups and Monoids: Homomorphism of Semigroups and Monoids, Subsemigroups and Submonoids; Groups: Definitions, Subgroups and Homomorphisms, Cosets and Lagrange’s Theorem, Normal Subgroups, algebraic Systems with Two Binary Operations; Group Codes: The Communication Model and Basic Notions of Error Correction, Hamming Distance.

Unit IV: Boolean Algebra **Hours: 7**

Lattices, Boolean Algebra: Boolean Functions, Representing Boolean Functions, sum of product expansions, Functional Completeness, Logic Gates, Combinations of Gate, Minimization of Circuits, Karnaugh Maps.

Unit V: Tree **Hours: 7**

Introduction, Rooted Tree, ordered rooted tree, tree as model, Properties of Trees, Applications of tree, Binary Search Trees, Decision Trees, Prefix Codes, Huffman Coding, Game Trees, Tree traversal, Preorder Traversing, Inorder Traversing, Post order Traversing, Spanning Tree, Minimum spanning tree

Unit VI: Graph **Hours: 7**

Graph Models; Basic Terminology, Special Simple Graphs, Bipartite Graphs, Matchings, Applications of Special Types of Graphs, New Graphs from Old; Graph Representation, Adjacency and Incidence

Matrices, Isomorphism of Graphs, Determining Isomorphism; Paths, Connectedness in Undirected Graphs and Directed Graphs, Paths and Isomorphism,
Counting Paths Between Vertices; Euler Paths and Circuits, Hamilton Paths and Circuits, Applications of Hamilton Circuits; Planar Graphs: Euler's Formula, Kuratowski's Theorem; Graph Coloring: Introduction, Applications of GraphColorings;

Text Books:

[1] Kenneth H. Rosen: Discrete Mathematics and Its Applications, 7th Edition, McGraw-Hill.

[2] J. P. Tremblay and R. Manohar: Discrete Mathematical Structures with Applications to Computer Science, Tata
McGraw-Hill Edition, McGraw-Hill.

Reference Books:

[1] Norman L. Biggs: Discrete Mathematics, 2nd Edition, Oxford University Press.

[2] Seymour Lipschutz and Marc Lars Lipson: Schaum's Outline of Theory and Problems of Discrete Mathematics, 3rd
Edition, Schaum's Outlines Series, McGraw-Hill.

[3] C. L. Liu and D. P. Mohapatra: Elements of Discrete Mathematics: A Computer Oriented Approach, 3rd Edition,
Tata McGraw-Hill, Mc-Graw Hill.

3KS03 OBJECT ORIENTED PROGRAMMING

Course Pre-requisite: Computer Programming

Course Objectives:

1. To explore the principles of Object Oriented Programming (OOP) such as data abstraction, encapsulation, inheritance and polymorphism.
2. To use the object-oriented paradigm in program design.
3. To Provide programming insight using OOP constructs.
4. To lay a foundation for advanced programming

Course Outcomes : On completion of the course, the students will be able to

1. Apply Object Oriented approach to design software.
2. Implement programs using classes and objects.
3. Specify the forms of inheritance and use them in programs.
4. Analyze polymorphic behaviour of objects.
5. Design and develop GUI programs.
6. Develop Applets for web applications

Unit I: Introduction to Object Oriented Programming (Hours:7)

Introduction, Need of OOP, Principles of Object-Oriented Languages, Procedural Language Vs OOP, Application of OOP, Java Virtual Machine, Java features, Program Structures. Java Programming Constructs: Variables, Primitive data types, Identifier, Literals, Operators, Expressions, Precedence Rules and Associativity, Primitive Type Conversion and Casting, ,Flow of Control.

Unit II: Classes and Objects (Hours:7)

Classes, Objects, Creating Objects, Methods, Constructors, Cleaning up Unused Objects, Class Variable and Methods, this keyword, Arrays, Command Line Arguments.

Unit III:Inheritance, Interfaces and Packages (Hours:6)

Inheritance: Inheritance vs. Aggregation, Method Overriding, super keyword, final keyword, Abstract class. Interfaces: Defining interfaces, Implementing interfaces, Accessing interface variables, Extending interfaces. Packages: Packages, java.lang package, Enum type.

Unit IV: Exception handling and Input /Output (Hours:7)

Exception: Introduction, Exception handling Techniques, User-defined exception, Exception Encapsulation and Enrichment. Input/Output: The java.io.file Class, Reading and Writing data, Randomly Accessing a file, Reading and Writing Files using I/O Package.

Unit V: Applets (Hours:7)

Introduction, Applet Class, Applet structure, Applet Life cycle, Common Methods used in displaying the output, paint (), update () and repaint (), More about applet tag, get Document Base() and get Code Base () methods, Applet Context Interface, Audio clip, Graphic Class, Color, Font, Font Metrics.

Unit VI: Unit Title: Event Handling (Hours:6)

Introduction, Event delegation Model, java.awt.event Description, Sources of events, Event Listeners, Adapter classes, Inner Classes. Abstract Window Toolkit: Introduction, Components and Containers, Button, Label, Checkbox, Radio Buttons, List Boxes, Choice Boxes, Textfield and Textarea, Container Class, Layouts, Menu, Scrollbar.

Text Books:

1. Sachin Malhotra and Saurabh Choudhary: Programming in Java, Oxford University Press 2010.
2. Herbert Schildt: Java Complete References (McGraw Hill)

Reference Books:

1. H.M.Dietel and P.J.Dietel, “Java How to Program” Pearson Education/PHI, Sixth Edition.
2. E. Balagurusamy: Programming with Java (McGraw Hill)
3. Dr. R. NageswaraRao: Core Java An Integrated Approach (Dreamtech)
4. Khalid Mughal: A Programmer’s Guide to Java Certification, 3rdEdition (Pearson)
5. Sharnam Shah and Vaishali Shah: Core Java for Beginners, (SPD), 2010.

3KS04 / 3KE04 DATA STRUCTURES

Course Pre-requisite: Fundamentals of programming Language &amp; Logic Building Skills.

Course Objectives:

1. To understand the linear and nonlinear data Structures and its memory representations.
2. To perform different operations on data structures such as insertion, deletion, searching and traversing.
3. To understand various data searching and sorting methods with its complexity.
4. To introduce various techniques for representation of the data in the real world.

Course Outcomes: On completion of the course, the students will be able to

1. Apply various linear and nonlinear data structures
2. Demonstrate operations like insertion, deletion, searching and traversing on various data structures
3. Examine the usage of various structures in approaching the problem solution.
4. Choose appropriate data structure for specified problem domain

Unit I: Introduction to Data Structures: (Hours:7)

Introduction to Data structures, Data Structure Operations, Algorithmic Notation, Complexity of algorithms. String processing: storing strings, character data type, string operations, word processing, and pattern matching algorithms.

Unit II: Array& Record Structure: (Hours7)

Linear arrays: Memory Representation of arrays, traversing linear arrays, insertion &amp; deletion operations, Bubble sort, Linear search and Binary search algorithms. Multi dimensional arrays, Pointer arrays. Record structures.

Unit III: Linked lists: (Hours7)

Linked lists: Memory Representation of Linked List, traversing a linked list, searching a linked list. Memory allocation &amp; garbage collection. Insertion & deletion operations on linked lists. Header linked lists, Two- way linked lists.

Unit IV: Stack & Queue: (Hours7)

Stacks: Sequential Memory Representation of Stack, Arithmetic expressions: Polish notation. Quick sort, Recursion, Tower of Hanoi. Queues: Sequential Memory Representation of Queue, DeQueue, Priority queues.

Unit V: Trees: (Hours7)

Introduction to Trees, Binary trees, Memory Representation of Binary Tree, Traversing binary trees, Header nodes, Binary Search Tree, Heap and heap sort, Path length &amp; Huffman’s algorithm.

Unit VI: Graphs & Sorting Algorithms: (Hours7)

Introduction to Graphs, Memory representation of graphs, Warshalls’ algorithm, operations on Graphs, Breadth First Search, Depth First Search. Sorting : Insertion Sort, Selection Sort, Radix sort, Merge Sort.

Text Books:

1. Seymour Lipschutz: Data Structures, Schaum’s Outline Series, McGraw-Hill, International Editions.
2. Trembley, Sorenson: An Introduction to Data Structures with Applications, McGraw Hill.

Reference Books:

1. Ellis Horowitz, Sartaj Sahni: Fundamentals of Data Structures, CBS Publications.
2. Data Structure Using C, Balagurusamy.
3. Standish: Data Structures in Java, Pearson Education.

3KS05 ANALOG & DIGITAL ELECTRONICS

Course Prerequisite: Basic Physics.

Course Objectives:

1. To get the introductory knowledge of PN Junction Diode, Bipolar Junction Transistor, Field Effect Transistor.
2. To understand number systems and conversion between different number systems.
3. To get basics knowledge about digital ICs and digital systems.
4. To study the design of combinational circuits and sequential circuits.

Course Outcomes:

At the end of course students will be able to:

1. Explain basic concepts of semiconductor devices and its application.
2. Compare different Number System and basics of conversion of number systems.
3. Realize different minimization techniques to obtain minimized expression.
4. Design Combinational Circuits.
5. Design and Develop Sequential Circuits.

Unit I: Diode and Characteristics: (Hours7)

PN-Junction Diode, Characteristics and Parameters, Zener Diode, Zener Diode as voltage regulator, Light Emitting Diode characteristics, Seven Segment Display, Photo Diode, PIN Diode

Unit II: Transistors and Characteristics: (Hours7)

Transistors and their Types (PNP, NPN), Transistor as an amplifier, BJT operation, BJT Voltages and Currents, BJT Switching, Common-Base Characteristics, Common-Emitter Characteristics, Common-Collector Characteristics, Transistor testing.

Unit III: Number System: (Hours7)

Binary Number System, Signed and unsigned Number, Octal Number System, Hexadecimal Number System,Conversions between Number Systems, r’s and (r-1)’s Complements Representation, Subtraction using 1’s and 2’s Complements, BCD, Gray Code, Excess 3 Code and Alpha numeric codes.

Unit IV: Minimization Techniques: (Hours7)

Logic Gates, Boolean Algebra, Logic Operation, Axioms and Laws of Boolean Algebra, Reducing Boolean Expression, Boolean Functions and their representation, SOP Form, POS Form, Karnaugh Map (up to 5 variable), Limitation of Karnaugh Map, Quine- McCluskey Minimization Technique (up to 5 variable).

Unit V: Combinational Circuits: (Hours7)

Introduction, Design Procedure, Adders, Subtractors, Binary Parallel Adder, 4 Bit Parallel Subtractor, Look-aheadcarry Adder, BCD adder, BCD Subtractor, Multiplexer, De-multiplexer, Decoder, Encoder, Comparator, Parity bit Generator/Checkers, Boolean Expression Implementation using these ICs.

Unit VI: Sequential Circuits: (Hours7)

Flip-flops: S-R, J-K, Master slave J-K, D-type, T-type, Flip flop Excitation Table, Conversion of Flip Flops, Registers: SISO, SIPO, PISO, PIPO, Universal Shift Register. Counters: Asynchronous and Synchronous counter, Up/Down counter, MOD-N counter, Ring counter, Johnson counter.

Text Books:

1. David A. Bell: "Electronic Devices and Circuits", 5e , Oxford University Press.
2. Jain R.P. "Modern Digital Electronics", 3e, TMH.

Reference Books:

1. Millman & Halkies: "Electronic Devices & Circuits", 2e, McGraw Hill.
2. Sedra& Smith: "Microelectronics Circuits", 5e, Oxford University Press.
3. Anand Kumar: "Switching Theory and Logic Design", 3e, PHI Learning Private Limited
4. Wakerly, "Digital Design: Principles and Practices", 3 e, Pearson Education, 2004.

3KS06 OBJECT ORIENTED PROGRAMMING - LAB

Course Pre-requisite: Basic Computer Programming

Course Objectives: Design, implement, test, and debug simple programs in an object-oriented programming
language.

1. To develop the knowledge of object-oriented paradigm in the Java programming language.
2. To evaluate classical problems using java programming.
3. To develop software development skills using java programming for real world applications.

Course Outcomes : On completion of the course, the students will be able to

1. Design, implement, test, and debug simple programs in an object-oriented programming language.
2. Interpret the basics of object-oriented design and the concepts of encapsulation, abstraction, inheritance, and polymorphism
3. Build applications in Java by applying concepts like interfaces, packages and exception handling.
4. Make use of Java concepts like API, Applets, AWT.

List of Experiments:

This is a sample list of Experiments; **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

1. Introduction to Object Oriented Programming and installation of JDK. Write a program to print a message "Hello World..."
2. Develop a program to explain use of Operators in java.
3. Develop a Program to study and implement Looping Statements belonging to Java**.**
4. Develop a Program to study and implement Selection Statements belonging to Java**.**
5. Develop a program to study and implement some Pyramid.
6. Develop a program to demonstrate the concept of Class, Method and Object.
7. Develop a program to study and implement the concept of Method Overloading.
8. Develop a program to study and implement concept of Constructor in Java.
9. Develop a program to study and implement concept of Constructor Overloading in Java.
10. Develop a program to study and implement the Array in Java.
11. Develop a Program on various ways to accept data through keyboard(Command Line Argument)
12. Develop a program to study and implement the concept of Inheritance.
13. Develop a program to study and implement the concept of Method Overriding.
14. Develop a program to study and implement the Abstract Class.
15. Develop a program to study and implement the concept of Interface in Java.
16. Develop a program to study and implement Exception Handling Mechanism in Java.
17. Develop a program to study and implement Java I/O.
18. Develop a program to study and implement simple Applet in java.

19. Develop a program on Applet to demonstrate Graphics, Font and Color class.
20. Develop a Program on passing parameters to applets
21. Develop a Program to create GUI application without event handling using AWT controls
22. Develop a Program to create GUI application with event handling using AWT controls
23. Develop a program on Multithreading
24. Develop a Program to create GUI application with event handling using Swing controls
25. Mini Project based on content of the syllabus. (Group of 2-3 students)

3KS07 DATA STRUCTURE - LAB

Course Pre-requisite: Basics of programming Language & Logic Building Skills

Course Objectives:

1. To understand the linear and nonlinear data Structures and its memory representations.
2. To perform different operations on data structures such as insertion, deletion, searching and traversing.
3. To understand various data searching and sorting methods with its complexity.
4. To introduce various techniques for representation of the data in the real world.

Course Outcomes : On completion of the course, the students will be able to

1. Apply various linear and nonlinear data structure.
2. Demonstrate operations like insertion, deletion, searching and traversing on various data structures
3. Examine the usage of various structures in approaching the problem solution.
4. Choose appropriate data structure for specified problem domain

List of Experiments:

This is a sample list of Experiments; **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

1. Write a program to find out largest number from the array and also find it’s location.
2. Write a program to traverse an array and find the sum and average of data elements from an array.
3. Write a Program to a) insert an element in an array b)delete an element from an array.
4. To study and execute the Linear search method
5. To study and execute the Binary Search method
6. To study and execute the Pattern matching Algorithms(Slow and Fast)
7. To study and execute Bubble sort method.
8. To study and implement various operations on singly linked list
(a) Traversing the linked list.
(b) Insert a node at the front of the linked list.
(c) Delete a last node of the linked list.
(d) Searching a Linked list.
9. To study and implement following operations on the doubly linked list.
(a) Insert a node at the front of the linked list.
(b) Insert a node at the end of the linked list.
(c) Delete a last node of the linked list.
(d) Delete a node before specified position.
10. To study and implement following operations on the circular linked list.
(a) Insert a node at the end of the linked list.
(b) Insert a node before specified position.
(c) Delete a first node of the linked list.
(d) Delete a node after specified position.
11. Understand the stack structure and execute the push, pop operation on it.
12. Understand the Queue structure and execute the insertion, deletion operation on it.
13. Formulate and demonstrate Transforming Infix Expressions to Postfix Expression using Stack.
14. Formulate and demonstrate the Evaluation of Postfix Expression using Stack.

15. To study and execute Quick sort method.
16. Understand the Tree structure and implement the Pre-order, In-order, post-order traversingoperations on it.
17. Understand the concept of Recursion and write a program to calculate factorial of a numberusing Recursion.
18. Understand the Heap sort and implement it on given data.
19. Understand the Insertion sort and implement it on given data.
20. Understand the Selection sort and implement it on given data.
21. To study and execute Merge sort method.
22. To study and execute Radix sort method.
23. Write a Program to implement the concept of BFS algorithm.
24. Write a Program to implement the concept of DFS algorithm.
25. To study and execute Josephus problem.

3KS08 ANALOG & DIGITAL ELECTRONICS - LAB

Course Pre-requisite: Students should have the knowledge of Basic Physics.

Course Objectives:

1. To impart the concepts of analog and digital electronics practically.
2. To provide students basic experimental experiences in the operation of semiconductor device and Digital ICs.
3. To learn the operation of various logic gates and their implementation using digital IC’s.
4. To learn the realization of various combinational and sequential circuits.

Course Outcomes : After successfully completing the lab, the students will be able to

1. Apply practically the concepts of analog and digital electronics.
2. Explain the operation and characteristics of semiconductor devices.
3. Illustrate the operation of various logic gates and their implementation using digital IC’s.
4. Design and implement various combinational logic circuits.
5. Design and implement various sequential logic circuits

List of Experiments:

This is a sample list of Experiments; **minimum 10 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

1. To study V-I characteristics of a PN Junction diode in Forward and Reverse bias.
2. To Sketch and Study the input and output characteristics of transistor connected in Common Emitter (CE) configuration..
3. To Sketch and Study the input and output characteristics of transistor connected in Common Base (CB) configuration
4. To Sketch and Study the input and output characteristics of transistor connected in Common Collector (CC) configuration.
5. To plot static characteristics of FET & calculate its parameters gm, rd and μ.
6. To implement Logic gates using TTL ICs (7400, 7402, 7404, 7408, 7410, 7411, 7420, 7427, 7432, 7486).
7. Study and verify the truth table of half adder and full adder using logic gates.
8. Study and verify the truth table of half subtractor and full subtractor using logic gates
9. To compare two 4 bits number and verify the output using 4-bit comparator IC 7485.
10. Implementation of 4×1 multiplexer using logic gates.
11. Implementation and verification of Demultiplexer and Encoder using logic gates.
12. Implementation of 4bit parallel adder using 7483 IC.
13. Design and verify the 4 bit synchronous counter.
14. Design and verify the 4 bit asynchronous counter.
15. Verification of truth table of SR, JK, T and D Flip Flops.

List of Experiments beyond syllabus:

1. Design and Implementation of Op-amp as an inverting amplifier.
2. Design and Implementation of Op-amp as a non-inverting amplifier.
3. To design and find frequency of A stable multi-vibrator using IC 555.

3KS09 C SKILL - LAB - I

Course Prerequisite: Basic knowledge of any Programming Language

Course Objectives:

1. To be able to program design with functions using Python.
2. To understand data and information processing techniques.
3. To understand to Design a program to solve the problems.
4. To be able to access database using python programming.
5. To be able to design web applications using python programming.

Course Outcomes : On completion of the course, the students will be able to

1. Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python
2. Interpret different Decision Making statements, Functions, Object oriented programming in Python
3. Summarize different File handling operations
4. Explain how to design GUI Applications in Python and evaluate different database operations
5. Develop applications using Django framework or Flask

List of Experiments:

This is a sample list of Experiments, **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

1. Write python program to store data in list and then try to print them.
2. Write python program to print list of numbers using range and for loop
3. Write python program to store strings in list and then print them.
4. Write python program in which an function is defined and calling that function prints Hello World.
5. Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST 2017”
6. Write a program to create, append, and remove lists in python.
7. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
8. Write a program to demonstrate working with tuples in python.
9. Write a program to demonstrate working with dictionaries in python.
10. Write a python program to find largest of three numbers.
11. Write python program in which an function(with single string parameter) is defined and calling that function prints the string parameters given to function.
12. Write python program in which an class is define, then create object of that class and call simple print function define in class.
13. Write a Python script that prints prime numbers less than 20.
14. Write a python program to find factorial of a number using Recursion.
15. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
16. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.
17. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
18. Write a Python class to convert an integer to a roman numeral.
19. Write a Python class to implement pow(x, n)
20. Write a Python class to reverse a string word by word.
21. Accessing and working with databases using Python.
22. Create data frame from .csv files and operations on it.
23. Plotting various graphs using Python.
24. Developing basic GUI using Python.

25. Developing web applications using Django framework or Flask

Reference Books :

1. “Core Python Programming”, R. NageswaraRao, dreamtech press.
2. “Python Programming A Modular Approach With Graphics, Database, Mobile and WebApplications”, SheetalTaneja, Naveen Kumar, Pearson.
3. Python Web Development with Django By Jeff Forcier, Paul Bissex, Wesley J Chun, Addison-Wesley Professional.
4. Kenneth A. Lambert, The Fundamentals of Python: First Programs, 2011, Cengage Learning
5. Allen B. Downey , “ Think Python: How to Think Like a Computer Scientist”, Second Edition, Shroff/O’Reilly Publishers
6. John V Guttag. “Introduction to Computation and Programming Using Python”, Prentice Hall of India
7. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, “Data Structures and Algorithms in Python”, Wiley
8. Introduction to Computation and Programming using Python, by John Guttag, PHI Publisher, Revised and Expanded version (Referred by MIT)

SEMESTER - IV

4KS01 ARTIFICIAL INTELLIGENCE

Course Pre-requisite: Basic concepts of Data Structures, Algorithms, Programming

Course Objectives:

1. To present an overview of Artificial Intelligence (AI) principles and approaches.
2. To understand the historical evolution of Artificial Intelligence.
3. To learn various searching techniques and identify to address a particular problem).

Course Outcomes : On completion of the course, the students will be able to

1. Explain concepts of Artificial Intelligence and different types of intelligent agents and their architecture.
2. Formulate problems as state space search problem & efficiently solve them.
3. Summarize the various searching techniques, constraint satisfaction problem and example problems - game playing techniques.
4. Apply AI techniques in applications which involve perception, reasoning and learning.
5. Compare the importance of knowledge, types of knowledge, issues related to knowledge acquisition and representation.

Unit I: Introduction to AI (Hours: 7)

Introduction : What Is AI?, The Foundations of Artificial Intelligence, The History of Artificial Intelligence, The State of the Art, Risks and Benefits of AI,

Intelligent Agents: Agents and Environments, Good Behavior: The Concept of Rationality, The Nature of Environments, The Structure of Agents

Unit II: Problem Solving Through AI (Hours: 7)

Introduction, Representation the AI Problems, Production System, Algorithm of Problem Solving, Examples of AI Problems, Nature of AI Problems

Unit III: Uninformed Search Strategies (Hours: 6)

Problem-Solving Agents, Example Problems, Search Algorithms, **Uninformed Search Strategies**: Breadth-First Search, Uniform-Cost Search, Depth First Search, Bidirectional Search, Depth Limited Search, Iterative Deepening Depth-First Search

Unit IV: Informed Search Strategies (Hours: 7)

Basic Concept of Heuristic Search and Knowledge, Designing of Heuristic Function, **Heuristic Search Strategies:** Generate-And-Test, Best-First Search, Problem Reduction, Hill Climbing, Constraint Satisfaction, Means-Ends-Analysis

Unit V: Adversarial Search & Games (Hours: 7)

Game Theory, Optimal Decisions in Games, Mini-Max Search, Alpha Beta Pruning, Additional Refinements, Monte Carlo Tree Search, Stochastic Games, Partially Observable Games, Limitations of Game Search Algorithms

Unit VI: Introduction to Knowledge (Hours: 6)

Introduction, Types of Knowledge, Knowledge Representation, Knowledge Storage, Knowledge Acquisition, Knowledge Organization and Management, Basic Concepts of Knowledge Engineering

Text Books:

1. Artificial Intelligence: A Modern Approach by Stuart Russell & Peter Norvig (Pearson - 4th Ed.)
2. Artificial Intelligence by Ela Kumar (IK International Publishing House Pvt. Ltd.)

Reference Books:

1. Artificial Intelligence by Elaine Rich and Kevin Knight (Tata McGraw Hill - 3rd Ed.)
2. A First Course in Artificial Intelligence by Deepak Khemani (Tata McGraw Hill - 1st Ed.)
3. Artificial Intelligence and Expert Systems by Patterson (PHI)
4. Introduction to Artificial Intelligence by RajendraAkerkar (PHI Learning Pvt. Ltd.)

4KS02 DATA COMMUNICATION AND NETWORKING

Course Prerequisite: Computer and Data Communication Requirements.

Course Objectives:

1. To understand the building blocks of digital communication system.
2. To prepare mathematical background for communication signal analysis.
3. To understand and analyze the signal flow in a digital communication system
4. To analyze error performance of a digital communication system in presence of noise and other interferences
5. To evaluate the errors using various error detection & correction techniques
6. To understand network based protocols in data communication and networking

Course Outcomes: On completion of the course, the students will be able to

1. Describe data communication Components Networks, Protocols and various topology based network architecture
2. Design and Test different encoding and modulating techniques to change digital to digital Conversion, analogto- digital conversion, digital to analog conversion, analog to analog conversion.
3. Explain the various multiplexing method and evaluate the different error detection & correction techniques
4. Illustrate and realize the data link control and data link protocols
5. Describe and demonstrate the various Local area networks and the IEEE standards

Unit I: Introduction to Data Communication (Hours:7)

Introduction Data Communication Components, Networks, Network types: Local Area Network, Wide Area Network, Switching. The Internet, Accessing the Internet, Standards and Administration: Internet Standard Internet Administration, Network Models TCP/IP Protocol Suite, the OSI Model.

Unit II: Guided &Unguided Media, Switching and Data Link Layer (Hours:7)

Transmission media introduction, Guided median & Unguided media-Wireless. Switching Introduction, Circuit Switch Networks Packet Switching Data Link Layer: Introduction, Nodes & Links, Services, Two categories of link, Two sublayers, Error detection and correction: Introduction, Block Coding, Cyclic codes, Checksum, Forward Error Correction, Data link control: DLC services, Data-Link Layer Protocol, HDLC, Point-To-Point Protocol, Media Access Control (MAC) Random Access, Controlled Access, Channelization.

Unit III: Network Layer (Hours: 7)

Introduction to Network layer Network Layer Services: Packetizing. Routing and Forwarding, Other Services Packet Switching: Datagram Approach: Connectionless Service, Virtual-Circuit Approach: Connection-Oriented Service, Network Layer performance: Delay, Throughput, Packet Loss, Congestion Control, IPV4 Address: Address Space, Classful Addressing. Classless Addressing, Dynamic Host Configuration Protocol (DHCP), Network Address Resolution (NAT), Forwarding of IP packets: Forwarding Based on Destination Address, Forwarding Based on Label, Routers as packet Switches

Unit IV: Network Layer Protocol (Hours: 7)

Network Layer Protocols: Internet Protocol (IP), Datagram Format, Fragmentation, Security of IPv4 Datagrams, ICMPV4: Messages, Debugging Tools, ICMP Checksum Mobile IP: Addressing. Agents, Three Phases, Inefficiency in Mobile IP Routing algorithms: Distance Vector routing, Link State Routing, IPV6 Addressing Representation, Address Space, Address Space Allocation, Auto configuration. Renumbering, Transition from IPV4 to IPV6: Strategies, Use of IP Addresses

Unit V: Transport Layer (Hours: 7)

Introduction to Transport layer: Introduction, Transport-Layer Services, Connectionless and Connection-Oriented Protocols, Transport-Layer Protocols: Simple Protocol, Stop-and-Wait Protocol, Go-Back-N Protocol (GBN). Selective- Repeat Protocol, Bidirectional Protocols: Piggy backing. User Datagram Protocols: User Datagram, UDP Services, UDP Applications, Transmission Control Protocol: TCP Services, TCP Features, Segment, A TCP Connection, State

Transition Diagram, Windows in TCP, Flow Control, Error Control, TCP Congestion Control, TCP Timers, Options, SCTP: SCTP Services, SCTP Features

Unit VI: Application layer (Hours: 7)

Introduction to Application layer: Providing Services. Application-Layer Paradigms, Client-Server Programming: Application Programming Interface, Using Services of the Transport Layer, Iterative Communication Using UDP, Iterative Communication Using TCP, Concurrent Communication, World wide web and HTTP World Wide Web,Hyper-Text Transfer Protocol (HTTP) FTP: Two Connections, Control Connection, Data Connection, Security for FTP,Electronic Mail: Architecture, Web-Based Mail, E-Mail Security, Domain Name System (DNS) Name Space, DNS in the Internet, Resolution, Caching, Resource Records, DNS Messages, Registrars, Security of DNS.

Text Book: Behrouz A. Forouzan: Data Communication and Networking. (5/e) (TMH).

Reference Books:

1. William Stallings: Data & Computer Communications, 6/e, Pearson Education
2. William L. Schweber: Data Communication, McGraw Hill
3. J.Freey: Computer Communication & Networks, AEW Press
4. D. Corner Computer Networks & Internet, Pearson Education.

4KS03 OPERATING SYSTEM

Course Pre-requisite: Discrete Structures, Data Structure, Any programming Language

Course Objectives:

1. To make students aware of the kernel and shell structure of the operating systems.
2. To make students aware of the purpose, structure and functions of operating systems
3. To equip students with understanding of the various scheduling algorithms in OS.
4. To make students aware of understanding of memory management in different OS.

Course Outcomes : On completion of the course, the students will be able to

1. Explain memory management issues like external fragmentation, internal fragmentation.
2. Illustrate multithreading and its significance.
3. List various protection and security mechanisms of OS.
4. Analyze and solve the scheduling algorithms.
5. Analyze the deadlock situation and resolve it.
6. Compare various types of operating systems

Unit I: Introduction to OS (Hours: 7)

Introduction: Operating System definition, OS Evolution, Components and Services, Process Concept, Process Scheduling, Operations on Processes, Cooperating Processes, Interprocess Communication, Threads Overview, Multithreading Models, Threading Issues, Java Threads

Unit II: Process Scheduling (Hours: 7)
Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, RR, Priority, Multilevel Queue, Multilevel Feedback Queue Scheduling

Unit III: Process Synchronization (Hours: 6)
Process Synchronization Basics: The Critical-Section Problem, Synchronization Hardware, Semaphores, Monitors, Deadlocks: Definition & Characterization, Deadlocks Prevention, Avoidance, Detection and Recovery from Deadlock

Unit IV: Memory Management (Hours: 7)
Memory Management Background, Swapping, Contiguous Memory Allocation Schemes, Paging, Segmentation, Virtual Memory Management: Background, Demand paging scheme, Process Creation, Page Replacement Policies, Allocation of Frames, Thrashing

Unit V: Unit Title: File System (Hours: 7)
File-System Interface; Directory Structure, File-System Mounting, File Sharing &Protection, File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management. File Recovery

Unit VI: Unit Title: I/O System (Hours:6)
I/O Systems : Overview, I/O Hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O to Hardware Operations , Disk Scheduling, Disk Management, Swap-Space Management, RAID Structure.

Text Book : Avi Silberschatz, P.B.Galvin, G.Gagne: “Operating System Concepts” (9/e) John-Wiley & Sons.

Reference Books:
1. A.S.Tanenbaum “Modern Operating Systems” Pearson Education.
2. William Stallings “Operating Systems” Prentice-Hall.
3. D. M. Dhamdhere “Operating Systems” Tata McGraw-Hill.
4. P. Balkrishna Prasad: “Operating Systems” Scitech Publications (I) Pvt. Ltd.

4KS04 MICROPROCESSOR & ASSEMBLY LANGUAGE PROGRAMMING

Course Pre-requisite: Computer Programming and Computer Fundamentals

Course Objectives:
1. To explore 8086 microprocessor and its architecture.
2. To introduce interfacing techniques of 8086 microprocessor.
3. To introduce basics of Internet of Things

Course Outcomes : On completion of the course, the students will be able to
1. Describe 8086 microprocessor and its architecture; also understand instruction processing during the fetch-decode-execute cycle.
2. Design and Test assembly language programs using 8086 microprocessor instruction set.
3. Demonstrate the implementation of standard programming constructs, including control structures and functions, in assembly language.
4. Illustrate and realize the Interfacing of memory & various I/O devices with 8086 microprocessor.
5. Explain the basic concepts of Internet of Things

Unit I: 8086 Architecture (Hours: 7)
8086 architecture and pin configuration, Software model of 8086 microprocessor. Memory addresses space and data organization. Data types. Segment registers, memory segmentation. IP & Data registers, Pointer, Index registers. Memory addresses generation.

Unit II: 8086 Instruction Set (Hours: 7)
8086 Instruction set overview, addressing modes. 8086 instruction formats. 8086 programming: Integer instructions and computations: Data transfer instructions, Arithmetic instructions and their use in 8086 programming.

Unit III: 8086 Instruction Set (Hours: 6)
8086 programming: logical instructions. Shift and rotate instructions and their use in 8086 programming. 8086 flag register and Flag control instructions, compare instruction, control flow and jump instructions, Loops & loop handling instructions. 8086 programming using these instructions.

Unit IV: Subroutines& Macros (Hours: 7)
The 8086 stack segment and stack related instructions. 8086 I/O Address space.Subroutines and related instructions, Parameter passing, Concept of Macros, Status saving on stack.Concept of recursion at assembly program level.8086 Programming using subroutines, recursion and macros.

Unit V: 8086 Interrupt (Hours: 7)
8086 Interrupts types, priority and instructions. Interrupt vector table, External hardware-interrupt interface signals & interrupts sequence. Software interrupts. Non-maskable interrupts. 8086 microprocessor interrupt programming.

Unit VI: Internet of Things (IoT) (Hours: 6)
Internet of things: An overview, IoT conceptual framework, IoT Architectural View, Technology behind IoT, Sources of IoT, M2M communication, Examples of IoT.

Text Book:

1. A. K. Ray & K. M. Bhurchandi: Advanced Microprocessors & Peripherals, Third Edition (TMH).
2. Raj Kamal: Internet of Things, Architecture and Design Principals, McGraw Hill Education (India) Private Limited

Reference Books:

1. W. A. Triebel& Avatar Singh: The 8088/8086 Microprocessors (4e) (PHI /Pearson Education)
2. Liu & Gibson: The 8088/8086 Microprocessor Architecture Programming and Interface (6/e) (PHI)

4KS05 THEORY OF COMPUTATION

Course Pre-requisite: Discrete Mathematics, Data Structures

Course Objectives:

1. To understand different automata theory and its operation.
2. To understand mathematical expressions for the formal languages
3. To study computing machines and comparing different types of computational models
4. To understand the fundamentals of problem decidability and Un-Decidability

Course Outcomes: On completion of the course, the students will be able to

1. To construct finite state machines to solve problems in computing.
2. To write regular expressions for the formal languages.
3. To construct and apply well defined rules for parsing techniques in compiler
4. To construct and analyze Push Down, Turing Machine for formal languages
5. To express the understanding of the Chomsky Hierarchy.
6. To express the understanding of the decidability and un-decidability problems.

Unit I: Finite State Machines: (Hours 8)
Alphabet, String, Formal and Natural Language, Operations, Definition and Design DFA (Deterministic Finite Automata), NFA (Non Deterministic Finite Automata), Equivalence of NFA and DFA: Conversion of NFA into DFA, Conversion of NFA with epsilon moves to NFA, Minimization Of DFA, Minimization of Finite Automata (Construction of Minimum Automaton).

Unit II: Finite Automata with output and Regular Expression: (Hours 8)
Definition and Construction of Moore and Mealy Machines, Inter-conversion between Moore and Mealy Machines. Definition and Identities of Regular Expressions, Construction of Regular Expression of the given Language, Construction of Language from the RE.

Unit III: Regular Language and Regular Grammar: (Hours 8)

Conversion of FA to RE using Arden‟s Theorem, Inter-conversion RE to FA, Pumping Lemma for RL, Closure properties of RLs(proofs not required), Regular grammar, Equivalence of RG (RLG and LLG) and FA.

Unit IV: Context Free Grammar and Languages: (Hours 8)

Introduction, Formal Definition of Grammar, Notations, Derivation Process: Leftmost Derivation, Rightmost Derivation, Derivation Trees, Construction of Context-Free Grammars and Languages, Pumping Lemma for CFL, Simplification of CFG, Normal Forms (CNF and GNF), Chomsky Hierarchy.

Unit V: Pushdown Automata: (Hours 8)

Introduction and Definition of PDA, Construction of PDA, Acceptance of CFL, Equivalence of CFL and PDA:Inter-conversion , Introduction of DCFL and DPDA, Enumeration of properties of CFL, Context Sensitive Language, Linear Bounded Automata.

Unit VI: Turing Machines and Decidability and Un-Decidability: (Hours 8)

Formal definition of a Turing Machine, Design of TM,Variants of Turing Machines: Multi-tape Turing machines, Universal Turing Machine. Decidability of Problems, Halting Problem of TM, Un-Decidability: Recursive enumerable language, Properties of recursive &amp; non-recursive enumerable languages.

Text Books:

1. Hopcraft H.E. & amp; Ullman J: Introduction to Automata Theory, Languages and Computation.
2. Peter Linz: An Introduction to Formal Languages and Automata.

Reference Books:

1. Rajesh K. Shukla: Theory of Computation, CENGAGE Learning, 2009.
2. K V N Sunitha and N Kalyani: Formal Languages and Automata Theory, McGraw Hill,2010
3. Lewis H.P. and Papadimition C.H.: Elements of Theory of Computation
4. Mishra &Chandrashekharan: Theory of Computation
5. C.K.Nagpal: Formal Languages and Automata Theory, Oxford University Press, 2011.
6. Vivek Kulkarni : Theory of Computation, OUP India, 2013.

4KS06 DATA COMMUNICATION & NETWORKING LAB

Course Pre-requisite: Computer and Data Communication Requirements

Course Objectives:

1. To understand the working principle of various communication protocols
2. To understand and analyze the signal flow in a digital communication system.
3. To analyze error performance of a digital communication system in presence of noise and other interferences.
4. To evaluate the errors using various error detection & correction techniques.
5. To understand network based protocols in data communication and networking.

Course Outcomes : On completion of the course, the students will be able to

1. Analyze performance of various communication protocols
2. Implement Configure various network protocols.
3. Compare IP Address classes of networks

List of Experiments:

This is a sample list of Experiments; **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

1. To study various LAN topologies and their creation using network devices, cables and computers. .
2. To connect the computers in Local Area Network.
3. Familiarization with Networking Components and devices: LAN Adapters, Hubs, Switches, Routers etc.
4. Write a program of bit stuffing used by Data Link Layer
5. Write a program to implement CRC(Cyclic Redundancy Check)

6. Write a program to implement Checksum
7. Write a program to implement Sliding window
8. Configure Internet connection and use IP-Config, PING / Tracer and Net stat utilities to debug the network issues.
9. Configuration of TCP/IP Protocols in Windows and Linux.
10. Transfer files between systems in LAN using FTP Configuration, install Print server in a LAN and share the printer in a network.
11. Write a C Program to determine if the IP Address is in Class A, B, C, D, or E
12. Write a C Program to translate Dotted Decimal IP Address into 32 Bit Address.
13. Configure Host IP, Subnet Mask and Default Gateway in a System in LAN(TCP/IP Configuration)

4KS07 OPERATING SYSTEM - LAB

Course Pre-requisite: Basic computer programming

Course Objectives:

1. To make students aware of the kernel and shell structure of the operating systems.
2. To make students aware of the purpose, structure and functions of operating systems
3. To equip students with understanding of the various scheduling algorithms in OS.
4. To make students aware of understanding of memory management in different OS.

Course Outcomes : On completion of the course, the students will be able to

1. Explain memory management issues like external fragmentation, internal fragmentation.
2. Illustrate multithreading and its significance.
3. List various protection and security mechanisms of OS.
4. Analyze and solve the scheduling algorithms.
5. Analyze the deadlock situation and resolve it.
6. Compare various types of operating systems

List of Experiments:

This is a sample list of Experiments, **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

1. To study Linux Operating System along with its installation.
2. To Study and Execute basic file commands and process related open source Ubuntu commands
a. Commands to view all executing, block and suspended process.
b. Command to check and change the priority of process CPU utilization for executing processes.
c. Commands to check for child process, sub-processes, process tree, abort & end process and all other basics commands related to processes
3. Write a program for multithreading using C.
4. To simulate First Come First Serve & Shortest Job First process scheduling algorithm
5. To simulate Shortest Job First process scheduling algorithm
6. To simulate Preemptive Shortest Job First process scheduling algorithm
7. To implement Round Robin Process scheduling Algorithm
8. To implement Priority Based Process scheduling Algorithm
9. To implement and analyze multi-level queue scheduling algorithm
10. To implement the following file allocation strategies.
11. To simulate paging technique of memory management.
12. To implement theFIFO page replacement policy
13. To implement the LRU page replacement policy
14. To implement the optimalpage replacement policy
15. To simulate producer-consumer problem using semaphores.
16. To implement Dining-Philosophers problem to deal with concurrency control mechanism.
17. To implement contiguous memory allocation strategies to detect fragmentation using: First Fit, Best Fit and Worst Fit.
18. To implement FCFS Disk Scheduling algorithm

19. To implement SCAN Disk Scheduling algorithm
20. To implement C-SCAN Disk Scheduling algorithm
21. To simulate Bankers algorithm for deadlock avoidance
22. To implement following memory management techniques Implement MVT and MFT where memory block size is 100 for 5 processes. Enter no. of blocks for each process and calculate internal fragmentation.
23. To simulate LFU page replacement algorithms
24. To simulate the Single level directory file organization techniques.
25. To Simulate bankers algorithm for Dead Lock Avoidance (Banker‘s Algorithm)

4KS08 MICROPROCESSOR & ASSEMBLY LANG. PROG. - LAB

Course Pre-requisite: Computer Programming, Number System

Course Objectives: In this lab student will learn about' Microprocessor and Interfacing' in regards to digital
computer, microprocessor architecture, programming with 8086 microprocessor and
different peripherals**.**

Course Outcomes On completion of the course, the students will be able to

1. Analyze the internal workings of the microprocessor
2. Design and develop programs in Assembly Language Programming
3. Describe 8086 microprocessor and its architecture; also understand instruction processing during the fetch-decode-execute cycle.
4. Design and Test assembly language programs using 8086 microprocessor instruction set.
5. Demonstrate the implementation of standard programming constructs, including control structures and functions, in assembly language
6. Illustrate and realize the Interfacing of memory & various I/O devices with 8086 microprocessor.

List of Experiments:

This is a sample list of Experiments; **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

1. Installation and Introduction of TASM Assembler.
2. Write a program for addition of two 8-bits numbers and two 16-bits numbers.
3. Write a program for subtraction of two 8-bits numbers and two 16-bits numbers.
4. Write a program for multiplication of two 8-bits numbers.
5. Write a program for division of two 8-bits numbers
6. Write a program to check whether a given number is even or odd.
7. Write a program to demonstrate Logical Group and Shift Rotate Instructions.
8. Write a program to check whether a given number is positive or negative.
9. Write a program to find greatest of two 8-bits signed &unsigned numbers.
10. Block Transfer Program
11. Write a program to find Factorial of a number using loop instruction.
12. Write a program to find cube of a given number using Subroutine.
13. Write a program to find square of a given number using Subroutine.
14. Write a program to find square of a given number using Macro.
15. Write a program to find whether the string is palindrome or not.
16. To convert BCD Number Program
17. Write a program to perform Reverse of the String
18. Write a program to transfer 10-bytes from one memory bank to another memory bank.
19. Program for sorting an array for 8086 microprocessor.
20. To write an assembly language program to arrange the given numbers in descending order.
21. Program for searching for a number/character in a string for 8086 microprocessor.

4KS09 C-SKILL-LAB II

Course Pre-requisite: Basic knowledge of scripting language, Programming language.Basic understanding of Electronic concepts.

Course Objectives: To develop an ability to design and implement static and dynamic website and to develop embedded systems with the help of Raspberry Pi/Ardino.

Course Outcomes : On completion of the course, a student will be able to

1. Develop client server program and web applications
2. Make use of project-based experience for web application development.
3. Create embedded systems using Raspberry Pi/Ardino

List of Experiments:

This is a sample list of experiments, **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

1. Introduction to PHP and configure it to work with Apache Web Server.
2. Design web pages for your college containing a description of the courses, departments, faculties, library etc, use href, list tags.
3. Create your class timetable using table tag.
4. Create user Student feedback form (use textbox, text area , checkbox, radio button, select box etc.)
5. Create your resume using HTML tags also experiment with colors, text , link , size and also other tags you studied.
6. Design a web page of your home town with an attractive background color, text color, an Image, font etc. (use internal CSS).
7. Develop a JavaScript to display today's date.
8. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient.
9. Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).
10. Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.
11. Write a PHP program to display a digital clock which displays the current time of the server.
12. Write the PHP programs to do the following: a. Implement simple calculator operations. b. Find the transpose of a matrix.
13. Write a PHP program to sort the student records which are stored in the database using selection sort.
14. Study and Install IDE of Arduino and different types of Arduino.
15. Write program using Arduino IDE for Blink LED.
16. Write Program for RGB LED using Arduino.
17. Study the Temperature sensor and write a Program for monitor temperature using Arduino.
18. Study and Implement RFID, NFC using Arduino. Study and implement MQTT protocol using Arduino.
19. Study and Configure Raspberry Pi.
20. WAP for LED blink using Raspberry Pi.
21. Study and Implement Zigbee Protocol using Arduino / Raspberry Pi.
22. Create Smart Plugs with Arduino and Raspberry Pi.
23. Interfacing digital sensors with raspberry pi.
24. Creating a webpage to control I-O devices, Reading data from sensor and passing to web page.
25. Implement a program to access Analog sensor via wifi with HTML Web server.

B.E. Sem. V (Computer Science & Engineering)

5KS01 DATABASE MANAGEMENT SYSTEMS
Course Prerequisite: Discrete Mathematics, Data Structures and Algorithm
Course Objectives:
Throughout the course, students will be expected to demonstrate their understanding of Database Management
Systems by being able to do each of the following:
1. To understand the fundamental concepts of database management system.
2. To learn database query languages.
3. To give systematic database design approaches covering conceptual design, logical design and an overview of
physical design.
4. To understand the query processing and optimization.
5. To learn basics of transaction management and concurrency control.

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to
1. Model, design and normalize databases for real life applications.
2. Discuss data models, conceptualize and depict a database system using ER diagram.
3. Query Database applications using Query Languages like SQL.
4. Design & develop transaction processing approach for relational databases.
5. Understand validation framework like integrity constraints, triggers and assertions.

Unit I: Introduction to DBMS: (Hours 8)
Database System Applications, Purpose of database systems, View of Data, Database Languages Database Architecture, Database Users and Administrators, Entity- Relationship Model, Constraints, Removing redundant attributes in Entity sets, E-R diagrams, Reduction to Relational Schemas, E-R design issues, Extended E-R Features.

Unit II: Relational Algebra, SQL: (Hours 8)
Relational Model: Structure of Relational Databases, Database schema, keys, schema diagram, relational query languages, relational operators, The Relational Algebra, Overview of SQL query language, SQL data definition, Basic Structure of SQL queries, Additional basic operations, Set Operations, Null Values, Aggregate Functions, Nested Subqueries, Modification of the Database Operations, Join expressions, Views.

Unit III: Relational Database Design: (Hours 8)
Integrity Constraints, SQL data types and schemas, Authorization, Triggers, Features of good relational designs, atomic domains and First Normal Form, decomposition using functional dependencies, Functional dependency
theory, Decomposition using multi-valued dependencies, More Normal Forms, Database Design Process.

Unit IV: Query Processing and Query Optimization: (Hours 8)
Query Processing: Overview, Measures of Query Cost, Selection Operation, Sorting, Join Operation (Nested Loop & Block Nested Loop Join), Evaluation of Expressions, Query Optimization: Overview, Transformation of Relational Expressions Estimating Statistics of Expression Results, , Choice of Evaluation Plans, Materialized Views.

Unit V: Transaction Management: (Hours 8)
Transaction Concept, Simple transaction model, Storage structure, Transaction Atomicity and Durability, transaction isolation, Serializability, transaction isolation and atomicity, transaction isolation levels, Implementation of Isolation levels, Transactions as SQL statements

Unit VI: Concurrency Control and recovery system: (Hours 8)

Lock-Based Protocols, Deadlock Handling, Multiple Granularities, Timestamp- Based Protocols, Validation-Based Protocols, Multi-version schemes, Recovery system :Failure classification, Storage , Recovery & Atomicity, Recovery algorithm, buffer management, Failure with loss of nonvolatile storage ,early lock release and logicalundo operations, , Remote Backup Systems

Text Book:

Abraham Silberschatz, Henry F. Korth, S. Sudarshan, DATABASE SYSTEM CONCEPTS, Sixth Edition, McGraw
Hill

Reference Books:

1. Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, McGraw-Hill
2. Shamkant B. Navathe, RamezElmasri, Database Systems, Pearson Higher Education
3. Garcia-Molina, Ullman, Widom: Database System Implementation, Pearson education.
4. S. K. Singh: Database Systems, Concepts, Design and Applications, Pearson Education.
5. G.K. Gupta: Database Management Systems, McGraw Hill.
6. Toledo and Cushman: Database Management Systems, (Schaum's Outlines)

5KS02 COMPILER DESIGN

Course Pre-requisite: Basic knowledge of Discrete Mathematics, Theory of Computation

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of Compiler Design by being
able to do each of the following:

- To learn concepts of programming language translation and phases of compiler design
- To understand the common forms of parsers.
- To study concept of syntax directed definition and translation scheme for the representation of language
- To illustrate the various optimization techniques for designing various optimizing compilers.

Course Outcomes (Expected Outcome):

On completion of the course, the students will be able to:

1. Describe the fundamentals of compiler and various phases of compilers.
2. Design and implement LL and LR parsers.
3. Solve various parsing techniques like SLR, CLR, LALR.
4. Examine the concept of Syntax-Directed Definition and translation.
5. Assess the concept of Intermediate-Code Generation and run-time environment.
6. Explain the concept of code generation and code optimization.

Unit I: Introduction to Compiler: (Hours 7)

Introduction to Compilers: Language Processor, The Structure of a Compiler. Lexical Analysis: The role of lexical analyzer, Input Buffering, Specification of tokens, Recognition of tokens, The lexical analyzer generator Lex, FiniteAutomata.

Unit II: Syntax Analysis: (Hours7)

Syntax Analysis: The role of the parser, Review of context free grammar for syntax analysis: Parse Tree and Derivation, Ambiguity in Grammar, Elimination of left recursion and left factoring. Top down parsing: recursive descent parsing, predictive parsers, Transition diagrams for predictive parsers, FIRST and FOLLOW, LL (1) Grammars, Construction of predictive parsing tables, Non recursive predictive parsing, Error recovery in predictiveparsing.

Unit III: Bottom up parsing: (Hours7)

Bottom up parsing: Handle pruning, Stack implementation of Shift Reduce Parsing, conflicts during shift reduce parsing Introduction to LR parsing: Simple LR, Items and the LR(0) Automation, The LR-Parsing algorithm,Construction of SLR parsing table, More powerful LR Parsers: canonical LR(1) Items, Constructing LR(1) sets of items and canonical LR(1) parsing tables, Constructing LALR parsing tables, The parser generator Yacc.

Unit IV: Syntax Directed Translation: (Hours7)

Syntax Directed Translation: Syntax directed definitions, Inherited and synthesized attributes, Evaluation orders of SDD's: Dependency Graphs, S-attributed definitions, L-attributed definition. Application of Syntax-Directed Translation: Construction of syntax trees. Syntax-directed Translation Schemes.

Unit V: Intermediate-Code Generation: (Hours7)

Intermediate-Code Generation: Variants of Syntax Trees: Directed Acyclic Graphs(DAG), Three Address Code. Run Time Environments: Storage Organization, Static versus Dynamic Storage Organization, Stack Allocation of Space: Activation trees, Activation Records, Calling Sequences, Variable- Length data on stack. Access to Nonlocal Data on the Stack. Heap Manager: The Memory Manager. Introduction to Garbage Collection: Design Goals for Garbage Collectors.

Unit VI: Code Generation: (Hours7)

Code Generation: Issues in Design of a Code generator, The Target Language, Address in the target code, Basic blocks and flow graphs. Optimization of Basic Blocks, Peephole Optimization and The Principal sources of Optimization.

Text Book:

Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman Compilers: "Principles, Techniques and Tools", Pearson
Education Second Edition.

Reference Books:

1. D. M. Dhamdhere, Compiler Construction—Principles and Practice, (2/e), Macmillan India.
2. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman Compilers: "Principles, Techniques and Tools", Pearson Education
(Low Price Edition).
3. Andrew Appel, Modern Compiler Implementation in C, Cambridge University press.
4. K C. Louden "Compiler Construction—Principles and Practice" India Edition, CENGAGE.
5. Bennett J.P., "Introduction to Compiling Techniques", 2/e (TMH).

5KS03 COMPUTER ARCHITECTURE & ORGANIZATION

Course Prerequisite: Microprocessor& Assembly Language Programming

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of Computer Architecture &
Organization by being able to do each of the following:

1. To discuss the basic concepts and structure of computers.
2. To solve concepts of arithmetic operations.
3. To understand addressing modes and memory organization.
4. To analyze conceptualize multitasking ability of a computer and pipelining
5. To explain IO communication

Course Outcomes (Expected Outcome):

On completion of the course, the students will be able to

1. Discuss the basic structure of a computer.
2. Understand the basic operation of CPU.
3. Compare and select various Memory and I/O devices as per requirement.
4. Solve the concepts of number representation and their operation.
5. Explain the concept of parallel processing and pipelining.

Unit I: Basic Structure of Computer: (Hours7)

Bus structures, Addressing Methods, and Machine Program Sequencing: Memory Locations, Addresses, Instruction and instruction sequencing, Addressing Modes, Basic I/O Operations.

Unit II: Memory Unit: (Hours7)

Basic Concepts, Memory Hierarchy, Semiconductor RAM Memories, Internal Organization of Memory Chips, Static Memories, Dynamic Memories, Read Only Memories, Speed, Size and Cost.

Unit III: Processing Unit: (Hours7)

Fundamental Concepts, Execution of a Complete Instruction, Hardwired Control, Performance Consideration, Microprogrammed Control, Microinstructions, Microprogram Sequencing.

Unit IV: I/O Organization: (Hours7)

Accessing I/O Devices, Interrupts, Enabling and Disabling Interrupts, Handling Multiple Devices, DMA, I/O Hardware, Standard I/O Interfaces: SCSI.

Unit V: Arithmetic: (Hours7)

Number Representations, Design of Fast Adders, Signed Addition and Subtraction, Multiplication of Positive Numbers, Booth Multiplier, Fast Multiplication, Integer Division, Floating Point Numbers and Operations.

Unit VI: Parallel Organization and Pipelining: (Hours7)

Parallel Processing, Array Processors, The Structure of General-Purpose Multiple Processors, Symmetric, Multiprocessors, Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards.

TextBook: Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, Fifth Edition Tata Mc-Graw Hill**.**

Reference Books:

1. William Stallings, “Computer Organization and Architecture: Designing for Performance”, Eighth Edition,
Pearson.
2. John P. Hayes, “Computer Architecture and Organization”, McGraw Hill Publication.
3. DA Patterson and JLHennessy,Computer Organization and Design, Morgan Kaufmann Publisher, 2nd edition
4. A.S. Tanenbaum, "Structured Computer Organization", PHI Publication

5KS04 COGNITIVE TECHNOLOGIES (L-3, T-0, C-3)

Course Prerequisite: Basic knowledge of Artificial Intelligence, Programming and Data Structures.

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of
Cognitive Technologies by being able to do each of the following:

· This course intends to introduce concept of cognitive technologies and important approaches of cognitive
technologies.

· Student will learn and analyze key concept of cognitive technologies.

· Students will gain an understanding of innovation concepts, terminology, current and future trends in cognitive technologies.

· Introduces students to IBM Watson platform, an artificially intelligent computer system capable of answering questions posed in natural language, developed in IBM’s Deep QA project.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to

1. Describe the Cognitive computing and principles of cognitive systems.
2. Identify role of Natural Language Processing in cognitive system.
3. Outline application of advanced analytics in cognitive computing.
4. Justify role of Cloud and Distributed Computing in Cognitive Computing.
5. Assess the process of building a Cognitive Application.
6. Identify the Emerging Areas and Future Applications of Cognitive Computing.

Unit I: Foundation of Cognitive Computing & Design Principle of Cognitive Systems Hours: 07

The Foundation of Cognitive Computing: Cognitive Computing as a New Generation, The Uses of Cognitive Systems, What Makes a System Cognitive, Gaining Insights from Data, Domains Where Cognitive Computing IsWell Suited, Artificial Intelligence as the Foundation of Cognitive Computing, Understanding Cognition, Two Systems of Judgment and Choice, Understanding Complex Relationships Between Systems, The Elements of aCognitive System, Infrastructure and Deployment Modalities. Design Principles for Cognitive Systems: Components of a Cognitive System, Building the Corpus, Bringing Data into the Cognitive System, Machine Learning, Hypotheses Generation and Scoring, Presentation and VisualizationServices.

Unit II: NLP and Big Data in Cognitive System Hours: 07

Natural Language Processing in Support of a Cognitive System: The Role of NLP in a Cognitive System, Semantic Web, Applying Natural Language Technologies to Business Problems.The Relationship Between Big Data and Cognitive Computing**:** Dealing with Human-Generated Data, Defining Big Data, The Architectural Foundation for Big Data, Analytical Data Warehouses, Hadoop, Data in Motion and Streaming Data, Integration of Big Data with Traditional Data.

Unit III: Knowledge Representation and Advance Analytics in Cognitive Computing Hours: 06

Representing Knowledge in Taxonomies and Ontologies: Representing Knowledge, Developing a Cognitive System, Defining Taxonomies and Ontologies, Explaining How to Represent Knowledge, Models for Knowledge Representation. Applying Advanced Analytics to Cognitive Computing: Advanced Analytics Is on a Path to Cognitive Computing, Key Capabilities in Advanced Analytics, Using Advanced Analytics to Create Value, Impact of Open Source Tools on Advanced Analytics.

Unit IV: Role of Cloud and Distributed Computing in Cognitive Computing Hours: 07

The Role of Cloud and Distributed Computing in Cognitive Computing: Leveraging Distributed Computing for Shared Resources, Why Cloud Services Are Fundamental to Cognitive Computing Systems, Characteristics of Cloud Computing, Cloud Computing Models, Delivery Models of the Cloud, Managing Workloads, Security and Governance, Data Integration and Management in the Cloud.

The Business Implications of Cognitive Computing: Preparing for Change, Advantages of New Disruptive Models, What Does Knowledge Mean to the Business?, The Difference with a Cognitive Systems Approach, Meshing Data Together Differently, Using Business Knowledge to Plan for the Future, Answering Business Questions in New Ways, Building Business Specific Solutions, Making Cognitive Computing a Reality, How a Cognitive Application Can Change a Market.

Unit V: IBM Watson and Process of Building a Cognitive Application Hours: 07

IBM’s Watson as a Cognitive System: Watson Defined, Advancing Research with a “Grand Challenge”, Preparing Watson for Jeopardy, Preparing Watson for Commercial Applications, The Components of DeepQA Architecture. The Process of Building a Cognitive Application: The Emerging Cognitive Platform, Defining the Objective, Defining the Domain, Understanding the Intended Users and Defining their Attributes, Defining Questions and Exploring Insights, Creating and Refining the Corpora, Training and Testing. Building a Cognitive Healthcare Application: Foundations of Cognitive Computing for Healthcare, Constituents in the Healthcare Ecosystem, Learning from Patterns in Healthcare Data, Building on a Foundation of Big Data Analytics, Cognitive Applications across the Healthcare Ecosystem, Starting with a Cognitive Application for Healthcare, Using Cognitive Applications to Improve Health and Wellness, to Enhance the Electronic Medical Record and to Improve Clinical Teaching.

Unit VI: Emerging Areas and Future Application Hours: 06

Smarter Cities: Cognitive Computing in Government: How Cities Have Operated, The Characteristics of a Smart City, The Rise of the Open Data Movement Will Fuel Cognitive Cities, The Internet of Everything and Smarter Cities, Understanding the Ownership and Value of Data, Smarter Approaches to Preventative Healthcare, Building a Smarter Transportation Infrastructure, Using Analytics to Close the Workforce Skills Gap, Creating a Cognitive Community Infrastructure, The Next Phase of Cognitive Cities. Emerging Cognitive Computing Areas: Characteristics of Ideal Markets for Cognitive, Computing Vertical Markets and Industries. Future Applications for Cognitive Computing: Requirements for the

Next Generation, Technical Advancements That Will Change the Future of Cognitive Computing, What the Future Will Look Like, Emerging Innovations.

Text Book:
Judith Hurwitz, Marcia Kaufman and Adrian Bowles, "Cognitive Computing and Big Data Analytics", publication John Wiley & Sons, Inc, 2015.

Reference Books:
1. José Luis Bermúdez, Cognitive Science: An Introduction to the Science of the Mind, publication Cambridge University Press, New York, Second Edition.
2. Jay Friedenberg and Gordon Silverman, Cognitive Science: An Introduction to the Study of Mind, Sage Publications, Inc. London, 2014.
3. Huimin Lu (Editor), Cognitive Internet of Things: Frameworks, Tools and Applications, Springer Nature Switzerland AG 2020.
4. Danish Contractor and Aaditya Telang (Editors), Applications of Cognitive Computing Systems and IBM Watson, 8th IBM Collaborative Academia Research Exchange, publication Springer Nature Singapore Pte Ltd., 2017.
5. S. Bird, E. Klein, E. Loper (2009), Natural Language Processing with Python, O' Reilly Media.

5KS04 (Prof. Elect. – I) (ii) DATA SCIENCE AND STATISTICS

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Data Science and Statistics by being able to do each of the following:
1. To understand the need of data science and Statistics
2. To understand the knowledge of statistics data analysis techniques utilized in business decision making.
3. To understand and apply the different data modeling strategies.
4. To apply the learned concept for skillful data management.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:
1. Explain basics and need of data science
2. Describe proficiency with statistical analysis of data.
3. Demonstrate loading of data and its manipulation for data science
4. Perform linear and multiple linear regression analysis.
5. Develop the ability to build and assess classification-based models
6. Evaluate outcomes and make decisions based on data.

Unit I: Introduction to Data Science: (Hours7)
Data Science: Relation with other fields, Relationship between Data Science and Information Science, Computational Thinking, Skills for Data Science, Tools for Data Science, Issues of Ethics, Bias, and Privacy in Data Science Data: Data types, Data Collection, Data Pre-processing.

Unit II: Techniques: (Hours7)
Techniques: Introduction, Data Analysis and Data Analytics, Descriptive Analysis, Diagnostic Analytics, Predictive Analytics, Prescriptive Analytics, Exploratory Analysis, Mechanistic Analysis, Regression.

Unit III: Python for Data Science (Hours7)
Python: Introduction, Getting Access to Python, Basic Examples, Control Structures, Statistics Essentials: Importing Data, Plotting the Data, Correlation, Linear Regression, Multiple Linear Regression.

Unit IV: Machine Learning for Data Science (Hours7)
Introduction: Machine Learning, Classification (Supervised Learning), Clustering (Unsupervised Learning), Regression,Gradient Descent

Unit V: Supervised Learning (Hours7)
Supervised Learning: Introduction, Logistic Regression, SoftMax Regression, Classification with KNN, Decision Tree, Random Forest, Naïve Bayes.

Unit VI: Unsupervised Learning (Hours7)

Unsupervised Learning: Introduction, Agglomerative Clustering, Divisive Clustering, Expectation Maximization (EM), Introduction to Reinforcement Learning.

Text Book:
Chirag Shah,"A Hands-on Introduction to Data Science“, Cambridge University Press (2020) ISBN:978-1-108-47244-9.

Reference Books:

1. Cathy O'Neil and Rachel Schutt: Doing Data Science, First Edition, 2014, O'reilly Publications, ISBN:978-1-449- 35865-5.
2. Data Mining: Concepts and Techniques By Jiawei Han, Jian Pei, Micheline Kamber
3. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani: An Introduction to Statistical Learningwith

Applications in R, First Edition, 2013, Springer-Verlag New York, ISBN: 978-1- 4614-7137-0.

5KS04 INTERNET OF THINGS [L-3, T-0, C-3]

Course Prerequisite: Basic knowledge of Internet and Microprocessor & Assembly Language Programming

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Internet of Things by being able to do each of the following:

· To learn and understand fundamental of IoT
· To study the design methodology and different IoT platform
· To understand usefulness of IoT for society
· To design and implement application of IoT using various sensor

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:

1. Understand the basics of IoT
2. Understand design methodology and platforms involved in IoT
3. Apply the knowledge to interface various sensors with IoT development
4. Design and Implement IoT system for real time application

Unit I: Hours: 6

Introduction to Internet of Things, Definition & Characteristics of IoT, Physical Design of IoT Logical Design of IoT, IoT Enabled Technologies like Wireless Sensor Networks, Cloud Computing, Big data analytics, Communication protocols, Embedded Systems, IoT Levels & Deployment Templates, Domain Specific IoTs: Home, Cities, Environment, Energy systems, Logistics, Agriculture, Health & Lifestyle.

Unit II: Hours: 7

IOT & M2M: Introduction, M2M, Difference between IoT and M2M, SDN and NFV for IoT, Software defined networks, network function virtualization, IoT Systems Management, Simple Network Management Protocol (SNMP) ,Limitations of SNMP, Network Operator Requirements, NETCONF, YANG, IoT Systems Management with NETCONF-YANG, NETOPEER.

Unit III: Hours: 7

IoT Platforms Design Methodology, Case Study on IoT System for Weather Monitoring, Motivation for Using Python, IoT Systems - Logical Design using Python ,Installing Python, Python Data Types & Data Structures, Control Flow, Functions, Modules, Packages, File Handling l, Date/Time Operations, Classes, Python Packages of Interest for IoT

Unit IV: (**Hours: 7)** IoT Physical Devices & Endpoints, Raspberry Pi, About the Board, Linux on Raspberry Pi, Raspberry Pi Interfaces serial, SPI, I2C, Programming Raspberry Pi with Python, Controlling LED with Raspberry Pi, Interfacing an LED and switch with Raspberry Pi, Interfacing Light Sensor with Raspberry Pi Other IoT Devices, pcDuino, BeagleBone Black, Cubieboard.

Unit V: Hours: 7

IoT Physical Servers & Cloud Offerings, Introduction to Cloud Storage Models & Communication APIs , WAMP - AutoBahn for IoT , Xively Cloud for IoT , Python Web Application Framework - Django , Designing a RESTful Web API , Amazon Web Services for ,SkyNet IoT Messaging Platform.

Unit VI: Hours: 7

Case Studies Illustrating IoT Design, Introduction, Home Automation: Smart Lighting, Home Intrusion detection, Cities: Smart parking, Environment: Weather Monitoring System, Weather reporting Bot, Air pollution monitoring, Forest fire detection, Agriculture: Smart Irrigation, Productivity Applications: IoT printer.

Text Book: Arshdeep Bahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, ISBN:0: 0996025510, 13: 978-0996025515.

Reference Books:

1. Fundamentals of Python, K.A.Lambert and B.L.Juneja, Cengage Learning, 2012.
2. David Hanes, IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, Cisco Press, ISBN-13: 978-1-58714-456-1, ISBN-10: 1-58714-456-5, 2017
3. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", 1st Edition, Academic Press, 2014

5KS04 PE-I: (iv) INTRODUCTION TO CYBER SECURITY

Course Prerequisite: Computer Programming, Data Structure, Data Communication & Networking.

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of Introduction to Cyber Security by being able to do each of the following:

- Understand basics of Cybercrime and Information Security.
- To familiarize various cyber threats, attacks, Cyber offenses.
- Understand Cybercrime on Mobile and Wireless devices.
- Understand tools and methods used in Cybercrime.
- Understand Access Control and Authentication.
- Understand Intrusion Detection and Prevention.

Course Outcomes (Expected Outcome):

After completion of this course, the students should be able to:

1. Know fundamentals of Cybercrimes and Cyber offenses
2. Realize the Cyber threats, attacks and Vulnerabilities.
3. Explore the industry practices and tools.
4. Comprehend the Access Control and Authentication Process.
5. Implement Intrusion Detection and Prevention.

Unit I: (Hours 7)

Introduction to Cybercrime: Introduction, Cybercrime, Cybercrime and Information Security, Classifications of Cybercrimes, Cybercrime: The Legal Perspectives, Cybercrimes: An Indian Perspective, Cybercrime and the Indian ITA 2000, A Global Perspective on Cybercrimes, Cybercrime Era.

Unit II: (Hours 7)

Cyber offenses: Introduction, Attacks, Social Engineering, Cyber stalking, Cyber cafe and Cybercrime, Botnets, Attack Vector, Cloud Computing.

Unit III: (Hours 7)

Cybercrime: Mobile and Wireless Devices Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit Cards Frauds in Mobile and Wireless Computing, Security Challenges posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication Service Security, Attacks on Mobile/Cell Phones, Mobile Devices: Security Implementations for Organizations, Organizational

Measures for Handling Mobile, Devices Related Security Issues Organizational Security Policies and Measures in Mobile Computing, Laptops.

Unit IV: (Hours 7)

Tools and Methods Used in Cybercrime: Introduction, Proxy Servers and Anonymizers, Phishing, Password Cracking, Key loggers and Spywares, Virus and Worms, Trojan Horses and Backdoors, Steganography, DoS and DDoS Attacks, SQL Injection, Buffer Overflow, Attacks on Wireless Networks.

Unit V: (Hours 7)

Access Control and Authorization: Definitions, Access Rights, Access Control Systems, Authorization, Types of Authorization Systems, Authorization Principles, Authorization Granularity, Web Access and Authorization. Authentication: Definition, Multiple Factors and Effectiveness of Authentication, Authentication Elements, Types of Authentications, Authentication Methods.

Unit VI: (Hours 7)

System Intrusion Detection and Prevention: Definition, Intrusion Detection, Intrusion Detection Systems (IDSs), Types of Intrusion Detection Systems, The Changing Nature of IDS Tools, Response to System Intrusion, Challenges to Intrusion Detection Systems, Implementing an Intrusion Detection System, Intrusion Prevention Systems (IPSs), Intrusion Detection Tools.

Text Books:

1. Nina Godbole, Sunit Belapure, “Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal.
2. Perspectives”, Wiley India Pvt Ltd, ISBN: 978-81- 265-21791,2013
3. Joseph Migga Kizza, “A Guide to Computer Network Security”, Springer 2009.

Reference Books:

1. V.K. Pachghare, “Cryptography and information Security”, PHI Learning Private Limited, Delhi India.
2. Nina Godbole, “Information Systems Security”, Wiley India, New Delhi
3. Kennetch J. Knapp, “Cyber Security & Global Information Assurance”, Information Science Publishing.
4. James Graham, Richard Howard, Ryan Olson, “Cyber Security Essentials” CRC Press.
5. Jeetendra Pande, “Introduction to Cyber Security” Uttarakhand Open University, 2017.

5KS06 DATABASE MANAGEMENT SYSTEMS LAB [P-2, C-1]

Course Prerequisite: Basic concept of programming, Basic concepts of data structures

Course Objectives:

· To study the ER model which provides a high level view of the issues in database design, to capture the semantics of realistic applications within the constraints of a data model.

· To study the primary data model (relational model) for commercial data processing applications.

· To study the standard structured query language and retrieve the information from the database in various ways.

· To study the integrity and security constraints of the database by enforcing constraints.

Course Outcomes (Expected Outcome) On completion of the course, the students will be able to

1. Design ER model for any kind of application.
2. Design and develop database.
3. Apply normalization.
4. Query the database.
5. Apply various integrity constraints
6. Build indices, views
7. Implement triggers, assertions

List of Experiments: This is the sample list of Experiments; minimum 12 experiments are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

Practical 1: To Study a Database Modeling Tool.

Study of Data Modeling Tools:

• Take a description of the enterprise, create its corresponding ER Diagram and build a database model using any modeling tool. The following basic features of the modeling should be covered while building the model:

• Logical / Physical Modeling

• Adding an entity / its attributes, relationships (all kinds of relationships viz., parent-child, foreign key references, one to many, many to many etc)

• Forward / reverse engineering

• Details of forward engineering / schema generation

• Steps to generate the schema

Practical 2: To Study and implement DDL Commands

Implement the model created in Practical 1, in any of the DBMS like Oracle, MySQL, or Microsoft SQL Server database software.

· Creating the proper tables

· Insert the data into it.

· Study Dropping and Altering the Tables. Study the cascaded deletes.

Practical 3: To Study and implement DML Commands-I

• SQL queries : Write and execute different SQL queries

• Execute Simple queries using SELECT, FROM, WHERE clauses,

• In Where clause use different predicates involving OR,AND, NOT

• Rename operation

• Tuple Variables

• Write SQL for various String operations (%,_,*)

• Match beginning with

• Match ending with

• Substring

• Match exactly n characters

• Match at least n characters

• Sort the output of the query using Order by

• Write SQL using Having

Practical 4 : To Study and implement DML Commands-II Write SQL queries and perform

• Set membership operations

• In, not in

• Some

• All

• Exists and not exists, Test for emptyness using exists, not exists

• Test for absence of duplicates.

• Nested queries

Practical 5. Study and implement aggregation functions.

· Write different queries using following Aggregate functions

· Min (minimum 3 SQL queries)

· Max (minimum 3 SQL queries)

· Avg (minimum 3 SQL queries)

· Sum (minimum 3 SQL queries)

· Count (minimum 3 SQL queries)

Practical 6: Write SQL to create Views and Indexes.

Practical 7: Write SQL to perform the modifications to the database

Practical 8 : PL /SQL

Practical 9 : Database Access Using Cursors

Write a trigger to find the names and cities of customers who have more than xyz in any account.

Practical 10 : Triggers

- Write a trigger for dealing with the overdrafts (set the account balance to zero, and creating a loan in the amount of the overdraft. Keep account number as loan number in the loan table)
- Write a trigger for dealing with blank cities (set the city field to null when it is blank)

Practical 11: Procedures, functions

- Write atleast 2 functions, and demonstrate its use
- Write atleast 2 procedures, and demonstrate its use

Practical 12 : Web Programming with PL/SQL. (Contents beyond Syllabus)

HTTP, A Simple Example, Printing HTML Tables., Passing Parameters, Processing HTML Forms., Multi-Valued Parameters.

Practical 13: Develop a JDBC Applications, Retrieve the information by connecting to the database using a host language (JAVA, C, C++) (Contents Beyond Syllabus)

Practical 14: Web Programming with Java Servlets. (Connecting to the database) (Contents beyond Syllabus)

A Simple Servlet., HTTP Servlet API Basics.,HTML Form Processing in Servlets.

Practical 15: PHP : Develop a simple application to access the database using PHP (Contents beyond Syllabus)

Study of Open Source NoSQL Databases

Based on the concepts covered in text create a Mini Project:

Suggested Topics:

i. Bank database (Given in Korth book)
ii. University Database (Given in Korth book)
iii. Airline Flight Information System.
iv. Library Database Application.
v. University Student Database.
vi. Video Chain Database.
vii. Banking Database.
viii. BiBTeX Database.
ix. Music Store Database.
x. Online Auctions Database.
xi. A Web Survey Management System.

Text Book: Korth, Sudarshan, Silberschatz, Database System Concept, Mc-Graw Hill Mysql Reference Manual (for Mysql database)

Reference Books: (may be 5 to 6)

1. Kevin Roebuck, “Storing and Managing Big Data - NoSQL, HADOOP and More”, Emereopty Limited,
ISBN: 1743045743, 9781743045749
2. Kristina Chodorow, Michael Dirolf, “MangoDB: The Definitive Guide” ,O‟Reilly Publications, ISBN: 978-1-449-34468-9.
3. Adam Fowler, “NoSQL For Dummies”, John Wiley & Sons, ISBN-1118905628
4. C J Date, “An Introduction to Database Systems”, Addison-Wesley, ISBN: 0201144719.

5KS07 COMPILER DESIGN – Lab [P-2, C-1]

Course Prerequisite: Basic knowledge of C Programming, Data Structures, Theory of Computation.

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Compiler Design by being able to do each of the following:

· Know the basic components of a Compiler.
· To implement Lexical Analyzer using Lex tool and Syntax Analyzer using Yaac Tool.
· To implement various parsing methods.
· To implement code optimization techniques .

Course Outcomes (Expected Outcome):
On completion of the course, the students will be able to
1. Identify the fundamentals of compiler and its phases.
2. Use the powerful compiler generation tools such as Lex and Yacc.
3. Write a lexical scanner, either from scratch or using Lex.
4. Develop program for solving parser problems.
5. Examine the various optimization techniques.

List of Experiments: This is the sample list of Experiments; minimum 12 experiments are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

List of Experiments based on Syllabus: (Maximum 20)

1. Design a lexical analyzer for given language and the lexical analyzer should ignore redundant spaces, tabs and new lines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value. Simulate the same in C language.
2. Write a C program to identify whether a given line is a comment or not.
3. Implement a C program to check parenthesis of regular expression is balanced or not.
4. Implement a C program to construct NFA from regular expression.
5. Implement a C program to simulate Deterministic Finite Automation (DFA) for a string which ending with ‘a’, ‘a*b+’, ‘abb’.
6. Write a C program to construct of DFA from NFA.
7. Implement a Lex program to verify the parenthesis of a given expression is balanced.
8. Implement a Lex program to recognize the token like Digit, Identifier & Delimiter.
9. Implement the Lexical Analyzer using JLex, flex or other lexical analyzer generating tools.
10. Implement a Lex program to a valid arithmetic expression and to recognize the identifier and operators present.
11. Implement a Lex program to count words, characters, lines, vowels and consonants from given input.
12. Implement a Lex program to check given number is positive negative or zero.
13. Implement a Lex program to generate string which is ending with zeros.
14. Implement LEX and Yacc tool to implement desk calculator.
15. Write a C program for constructing of SLR parsing.
16. Write a C program for constructing of LL (1) parsing.
17. Write a C program for constructing of LALR parsing.
18. Write a C program for constructing recursive descent parsing.
19. Write a C program to implement Program semantic rules to calculate the expression that takes an expression with digits, + and * and computes the value.
20. Write a C program for Tokenizing the file which reads a source code in C/C++ from an unformatted file and extract various types of tokens from it
21. Write functions to find FIRST and FOLLOW of all the variables / given grammar.
22. Implement a Shift Reduce Parser for the following productions.
23. E → E+E / E*E / a / b
24. Implement a symbol table containing functions create(), modify(), search(), display() and delete().
25. Implement three address Code for the input a=b*c.
26. Implement Recursive Decent Parser for the productions.

List of Experiments beyond Syllabus: (Maximum 05)

1. Convert the BNF rules into Yacc form and write code to generate Abstract Syntax Tree.
2. Write a C program to generate machine code from abstract syntax tree generated by the parser.
3. Write a Lex program to find out total number of vowels, and consonants from the given input string.
4. Implementation of Finite State machines DFA, NFAs .
5. Computation of Leading & Trailing Sets.

Text Book: Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman Compilers: "Principles, Techniques and Tools", Pearson Education, Second Edition.

Reference Books:
1. Doug Brown, John Levine, and Tony Mason, "Lex & Yacc", O'Reilly & Associates, Inc., Second Edition.
2. Andrew Appel, "Modern Compiler Implementation in C", Cambridge University press.
3. K C. Louden "Compiler Construction - Principles and Practice" India Edition, CENGAGE.
4. Dick Grune, Kees van Reeuwijk, Henri E. Bal, Ceriel J.H. Jacobs and Koen Langendoen, "Modern Compiler Design", Second Edition, John Wiley & Sons Publication.
5. Keith Cooper and Linda Torczon, "Engineering: A Compiler", Second Edition, Morgan Kaufmann Publication.

5KS09 C-Skill Lab – III [P-2, C-1]
Course Prerequisite: Basic knowledge of Web Development, HTML, CSS, JavaScript and IDE.
Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of C-Skill
Lab - III by being able to do each of the following:
· To develop an ability to set up a local JS Library/Framework development Environment.
· To be able to install and implement different JS Libraries and Frameworks
· To be able to develop single-page/multi-page static and dynamic Web Applications.
Course Outcomes (Expected Outcome): On completion of the course, the students will be able to
1. Explain the various tools, packages and modules required for Web Development.
2. Discuss the workings of web server, cookies, routes, etc.
3. Develop a mobile application using JS Framework.
4. Design GUI using JS framework and/or Libraries.
5. Create applications using Angular, React, Node and Express.
List of Experiments: This is the sample list of Experiments; minimum 12 experiments are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

List of Experiments based on Syllabus: (Maximum 20)
1. Introduction to the Node.js and its installation to print Hello World
2. To study built-in modules and implement the user defined built-in modules in the Node.js
3. To study HTTP module and implement Node.js as a web server
4. To study and implement Node.js File system module to read, write, create, update, delete and rename the file
5. To study the URL module of the Node.js and write a program that opens the requested file and returns the content of the file to the client. If anything goes wrong, throw a 404 error.
6. To convert the output "Hello World!" into upper-case letters by installing the "upper-case" package of NPM.
7. To study event handling in Node.js and demonstrate it using event module and EventEmitter object.
8. To study and implement the Formidable module of Node.js to upload the file on the server.
9. To study and implement the Nodemailer module of Node.js to send emails from your server.
10. To install MySQL and its driver and create connection with it using Node.js.
11. To demonstrate the creation database and table in MySQL using Node.js
12. To demonstrate the insertion of single and multiple records in the MySQL using "INSERT" statement and Node.js
13. To demonstrate the display of records from the MySQL database using "SELECT" statement and display it using Node.js

14. To demonstrate the display the records based on condition from the MySQL database using “WHERE” statement using Node.js
15. To demonstrate deletion of records from database using “DELETE” statement and Node.js
16. To demonstrate updating existing records in a table by using the "UPDATE" statement and Node.js
17. To demonstrate combining rows from two or more tables, based on a related column between them, by using a JOIN statement using Node.js

List of Experiments beyond Syllabus: (Maximum 05)

1. Create an Email sender app using Node.js
2. Create an Basic User database: Site in which User can Sign up/Login and can see other User’s Profile Information.
3. Create a User model covering Registration, Email verification(send an email), login (with remember me, display user details and allow to save/update user details(DOB, Location, Hobbies etc or anything)
4. A random number generator web application.

Text Books:

1. Simon Holmes: Getting Mean with Mongo, Express, Angular, and Node, 2nd Edition, Manning.
2. Alex Banks and Eve Porcello: Learning React: Functional Web Development with React and Redux, O’Reilly .

Reference Books:

1. ShyamSeshadri: Angular Up and Running, O’Reilly
2. Akshat Paul and Abhishek Nalwaya: React Native for Mobile development, Apress.
3. Jos Dirksen: Learn Three.js, 3rd Edition, Packt Publishing.
4. Patrick Mulder and Kelsey Breseman: Node.js for Embedded Systems, O’Reilly

DRAFT

5KS08 EMERGING TECHNOLOGY LAB I

5KS08 Emerging Technology Lab 1 is based on 5KS04 Professional Elective-I. Tentative FOSS Tools & Technology for Practical’s are as follows:

AI : IBM Watson, Microsoft Cognitive Toolkit , TensorFlow, Apache SystemML, Caffe, OpenNN, Torch, Neuroph

DS :R, Python, Cassandra, Apache Hadoop,

IoT : Arduino, DeviceHive, Kaa, Home Assistant

Cyber Security: Kali Linux, OpenVPN, NMAP, Metasploit Framework

5KS08 EMERGING TECHNOLOGY LAB - I
(Data Science and Statistics Lab - I)

Course Prerequisite: Basic knowledge of Mathematics.

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of Data Science and Statistics by being able to do each of the following:

- Demonstrate knowledge of statistical data analysis techniques utilized in business decision making.
- Apply principles of Data Science to the analysis of business problems.
- Apply the learned concepts for the skillful data management.

Course Outcomes (Expected Outcome):

On completion of the course, the students will be able to:

1. Demonstrate proficiency with statistical analysis of data.
2. Build skills in transformation and merging of data for use in analytic tools.
3. Perform linear and multiple linear regression analysis.
4. Develop the ability to build and assess data-based models.
5. Evaluate outcomes and make decisions based on data.

List of Experiments:

This is the sample list of Experiments; minimum 12 experiments are to be performed covering the entire syllabus. At
least two (2) experiments should be beyond syllabi based on learning of syllabi.

List of Experiments based on Syllabus:

[1] Introduction to Basics of Python:

[2] Variable creation, Arithmetic and logical operators, Data types and associated operations Sequence data types and associated operations, Strings, Lists, Arrays, Tuples, Dictionary, Sets, Range, Loops and Conditional Statement's Linear Regression:

[3] To learn different Libraries in Python and to perform Simple Linear Regression and Multiple Linear Regression

[4] To learn Interaction Terms and to perform Non-linear Transformations of the Predictors

[5] To learn and evaluate Qualitative Predictors

[6] To learn to Write Functions Logistic Regression, LDA, QDA, and KNN

[7] To perform Logistic Regression

[8] To perform Linear Discriminate Analysis

[9] To perform Quadratic Discriminate Analysis

[10] To implement K-Nearest Neighbors technique

[11] To use Caravan Insurance Data for LR, LDA, QDA, and KNN Cross-Validation and the Bootstrap

[12] To learn and perform The Validation Set Approach

[13] To learn and perform Leave-One-Out Cross-Validation

[14] To learn and perform k-Fold Cross-Validation

[15] To learn and perform The Bootstrap Subset Selection Methods

[16] To learn and perform Best Subset Selection

[17] To learn and perform Forward and Backward Stepwise Selection

[18] To learn to Choose Among Models Using the Validation Set Approach and Cross-Validation Ridge Regression andthe Lasso

[19] To learn and perform Ridge Regression

[20] To learn and perform The Lasso Decision Trees

[26] To learn and perform Fitting Classification Trees

[27] To learn and perform Fitting Regression Trees

[28] To learn and implement Bagging and Random Forests

[29] To learn and perform Boosting. Support Vector Machines

[30] To learn and perform Support Vector Classifier

[31] To learn and perform Support Vector Machine

[32] To learn and perform ROC Curves

[33] To learn and perform SVM with Multiple Classes

[34] To use Gene Expression Data Clustering

[35] To implement K-Means Clustering

[36] To implement Hierarchical Clustering

[37] NCI60 Data Example

[38] To implement PCA on the NCI60 Data

[39] To Cluster the Observations of the NCI60 Data.

List of Experiments beyond Syllabus: (Maximum 05)

1. To implement the Association Rules
2. To implement the kernel method to increase data separation
3. Develop a data model and deploy it as R HTTP Services or by export
4. Develop a data model and present it to end user with proper presentations
5. Carry out your assigned task and present it to other data scientist with proper presentations.

Text Books:

1. Cathy O'Neil and Rachel Schutt: Doing Data Science, First Edition, 2014, O'reilly Publications, ISBN: 978-1-449- 35865-5

2. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani: An Introduction to Statistical Learning with Applications in R, First Edition, 2013, Springer-Verlag New York, ISBN: 978-1- 4614-7137-0.

Reference Book:

Nina Zumel, John Mount: Practical Data Science with R, First Edition, 2014, Manning Publications Co., ISBN: 9781617291562.

B.E. Sem. VI (COMPUTER SCIENCE & ENGINEERING)

6KS01 SECURITY POLICY & GOVERNANCE [L-3, T-0, C-3]

Course Prerequisite: Data Communication and Networking,

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Security Policy & Governance by being able to do each of the following:

1. Understand the legal and regulatory environment and its relationship to Information Security.
2. Understand Information Security Concepts.
3. Understand the role of Information Security governance and planning within the organizational context.
4. Understand how to develop, implement and maintain various types of Information Security policies.
5. Understand risk management and its role in the organization.
6. Understand how to identify risk control classification categories

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to

1. List and discuss the key characteristics of Information Security, Leadership and Management
2. Differentiate between Law and Ethics
3. Describe why ethical codes of conduct are important to Information Security
4. Discuss the importance, benefits and desired outcomes of Information Security Governance
5. Discuss the process of developing, implementing and maintaining various types of Information Security Policies.
6. Define Risk Management and its role in the organization.

Unit I: Hours:6

Introduction to the Management of Information Security: Introduction to Security, Key Concepts of Information Security: Threats and Attacks, Management and Leadership, Principles of Information Security Management.

Unit II: Hours:6

Compliance: Law and Ethics: Introduction to Law and Ethics, Ethics in information Security, Professional Organizations and Their Codes of Conduct, Information Security and Law Organizational Liability and the Management of Digital Forensics.

Unit III: Hours:6

Governance and Strategic Planning for Security: The Role of Planning, Strategic Planning, Information Security Governance, Planning for Information Security Implementation.

Unit IV: Hours:6

Information Security Policy: Policy, Enterprise Information Security Policy, Issue-Specific Security Policy, System- Specific Security Policy, Guidelines for Effective Policy Development and Implementation.

Unit V: Hours:6

Risk Management: Assessing Risk: Introduction to the Management of Risk in Information Security, The Risk Management Process.

Unit VI: Hours:6

Risk Management: Treating Risk: Introduction to Risk Treatment, Managing Risk, Alternative Risk Management Methodologies.

Text Book: Michael E. Whitman, Herbert J. Mofford, “Management of Information Security” Sixth Edition, Cengage Learning, 2016.

Reference Books:

[1] Robert F Smallwood, “Information Governance for Business Documents and Records” Wiley 2014

[2] Michael E. Whitman and Herbert J. Mofford, “Principles of Information Security” Sixth Edition, Cengage Learning, 2018

[3] Krag Brotby, “Information Security Governance: A Practical Development and Implementation Approach” 2009 by John Wiley & Sons.

[4] Brijendra Singh, “Network Security and Management” Second Edition, PHI.

[5] Alan Calder and Steve Watkins, “IT Governance an international guide to data security and ISO27001/ISO27002” 2015, Kogan Page Limited.

[6] Evan Wheeler, “Security Risk Management, Building an Information Security Risk Management Program from the Ground Up” 2011, Syngress publications.

[7] Mike Chapple, James Michael Stewart and Darril Gibson, “CISSP® Certified Information Systems Security Professional Official Study Guide” Eighth Edition, 2018, John Wiley & Sons.

6KS02 DESIGN AND ANALYSIS OF ALGORITHMS [L-4, T-0, C-4]

Course Prerequisite: Any programming language, Discrete Mathematics and Data Structures.

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Design and Analysis of Algorithms by being able to do each of the following:

1. To understand asymptotic analysis of algorithms.
2. To apply algorithmic strategies while solving problems.
3. Ability to analyze time and space complexity.
4. Demonstrate a familiarity with major algorithms.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to

1. Carry out the analysis of various Algorithms for mainly Time complexity.
2. Apply design principles and concepts to algorithm design.
3. Understand different algorithmic design strategies.
4. Analyze the efficiency of algorithms using time complexity.
5. Apply the standard sorting algorithms.

Unit I: Iterative Algorithm Design Issue: Hours: 8

Introduction, Use of Loops, Efficiency of Algorithms, Estimating & Specifying Execution Times, Order Notations,Algorithm Strategies, Design using Recursion

Unit II: Divide And Conquer Hours: 8

Introduction, Multiplication Algorithm and its analysis, Introduction to Triangulation, Covex Hulls, Drawbacks of D & C & Timing Analysis.

Unit III: Greedy Methods Hours: 8

Introduction, Knapsack Problem, Job sequencing with deadlines, Minimum Spanning Trees, Prim’s Algorithms, Kruskal’s Algorithm, Dijkstras Shortest Path Algorithm.

Unit IV: Dynamic Programming Hours: 8

Introduction, Multistage Graphs, Traveling Salesman, Matrix multiplication, Longest Common Sub-Sequences, Optimal Polygon Triangulation, Single Source Shortest Paths.

Unit V: Backtracking Hours: 8

Combinational Search, Search & Traversal, Backtracking Strategy, Backtracking Framework, and Some typical State Spaces.

Unit VI: Efficiency of Algorithm Hours: 8

Polynomial Time & Non Polynomial Time Algorithms, Worst and Average case Behavior, Time Analysis of Algorithm, Efficiency of Recursion, Complexity, Examples of Complexity Calculation for Various Sorting algorithms. Time-Space Trade off and Time-Space Trade off in algorithm research.

Text Book: Dave and Dave: "Design and Analysis of Algorithms" Pearson Education.

Reference Books:

[1] Aho,Hopcroft & Ullman "The Design & Analysis of Computer Algorithms", Addison-Wesley
[2] G. Brassard, P.Bratley: "Fundamentals of Algorithmics" , PHI
[3] Horowitz & Sahani: " Fundamental Algorithms" , Galgotia.
[4] Cormen, T.H, Lierson & Rivest: " Introduction to Algorithms", Mc Graw-Hill.

DRAFT

6KS03 SOFTWARE ENGINEERING [L-3, T-0, C-3]

Course Prerequisite: Fundamentals of Programming Languages.

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Software Engineering by being able to do each of the following:

1. To learn and understand the principles of Software Engineering
2. To be acquainted with methods of capturing, specifying, visualizing and analyzing software requirements.
3. To apply Design and Testing principles to S/W project development.
4. To understand project management through life cycle of the project.
5. To understand software quality attributes.
6. To understand of the role of project management including planning, scheduling, risk management.

Course Outcomes (Expected Outcome): On completion of the course, student will be able to–

1. Decide on a process model for a developing a software project
2. Classify software applications and identify unique features of various domains
3. Design test cases of a software system.
4. Understand basics of Project management.
5. Plan, schedule and execute a project considering the risk management.
6. Apply quality attributes in software development life cycle.
7. Understand quality control and to ensure good quality software.

Unit I: Introduction to Software Engineering, Software Process Models Hours: 6

Evolving role of Software, Software crises & myths, Software engineering, Software process & process models, Linear sequential, prototyping ,RAD ,Evolutionary Product & Process, Project management concepts, People, Product, Process, Project W5HH principles, critical practice

Unit II: Project Management: Process, Metrics, And Estimations & Risks Hours:6

Measures, Metrics & Indicators. Metrics in process & project domains-software measurement, Metrics for software quality, small organization. Software projects Planning: Scope, resources, estimation, decomposition technique, Tools. Software risks: identification, risk projection, refinement & RMMM plan

Unit III: Project Scheduling & Quality Management Hours: 6

Project Scheduling: Concepts. Peoples Efforts. Task set, Task network. Scheduling. EV analysis, Project Plan. Software quality concepts. SQ Assurance, Software reviews, technical reviews, software reliability, ISO 900 L, SQA Plan. SCM process. Version control. SCM standard.

Unit IV: Requirement Engineering & System Engineering Hours:6

System engineering: Hierarchy, Business Process & Product engineering: Overviews. Requirement engineering, System modeling. Requirement analysis. Analysis principles. Software prototyping. Specification. Design Process. Design Principles & Concepts. Effective modular design. Design model & documentation.

Unit V: Software architecture & User interface design Hours: 6

Software architecture, Data Design, Architectural styles, Requirement mapping. Transform & Transaction mappings. User interface design: Golden Rule. UTD, Task analysis & modeling, ID activities, Tools, design evaluation. Component level design: Structure programming, Comparison of design notation.

Unit VI: Software Testing Hours: 6

Software testing fundamentals; test case design, White box testing. Basis path, control structure-, Black box-Testing, & for specialized environments. Strategic approach to S/W testing. Unit testing, integration testing, validation testing, and system testing. Debugging. Technical metrics for software.

Text Book: Pressman Roger. S: Software Engineering, A Practitioner's Approach, TMH.

Reference Books:

[1] Somerville: Software Engineering (Addison-Wesley) (5/e)
[2] Fairly R: Software Engineering (McGraw Hill)
[3] Davis A: Principles of Software Development (McGraw Hill)
[4] Shooman, M.L: Software Engineering (McGraw-Hill)

6KS04 NATURAL LANGUAGE PROCESSING [L-3, T-0, C-3]

Course Prerequisite: Fundamentals of Artificial Intelligence.

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Natural Language Processing by being able to do each of the following:

1. To learn the fundamentals of natural language processing
2. To understand the use of CFG and PCFG in NLP
3. To understand the role of semantics of sentences and pragmatics
4. To gain knowledge in Information Extraction.

Course Outcomes (Expected Outcome): On completion of the course, student will be able to–

1. Understand how to tag a given text with basic Language features
2. Design an innovative application using NLP components
3. Implement a rule-based system to tackle morphology/syntax of a language
4. Design a tag set to be used for statistical processing for real-time applications
5. Compare and contrast the use of different statistical approaches for different types of NLP applications.

Unit I: Overview and Morphology Hours: 6

Introduction, Models and Algorithms, Regular Expressions Basic Regular Expression Patterns, Finite State Automata, Morphology, Inflectional Morphology, Derivational Morphology, Finite-State Morphological Parsing

Unit II: Word Level Analysis Hours: 6

Role of language models. Simple N-gram models. Estimating parameters and smoothing. Evaluating language models. Part Of Speech Tagging and Sequence Labeling Lexical syntax. Hidden Markov Models. Maximum Entropy models.

Unit III: Syntactic Analysis Hours: 6

Context-Free Grammars, Grammar rules for English, Treebanks, and Normal Forms for grammar, Dependency Grammar, Syntactic Parsing, Ambiguity, Probabilistic CFG, and Probabilistic Lexicalized CFGs.

Unit IV: Semantic Analysis Hours: 6

Representing Meaning, Meaning Structure of Languages, First Order Predicate Calculus, Syntax-Driven Semantic Analysis, Semantic Attachments, Syntax-Driven Analyzer, Robust Analysis, Relations among Lexemes and their Senses, Word Sense Disambiguation

Unit V: Learning to Classify Text: Hours: 6

Supervised classification, further examples of supervised classification, Evaluation, Decision Trees, Naïve Bayes classifiers, Modelling Linguistic Patterns.

Unit VI: Extraction Information from Text: Hours: 6

Information Extraction, Chunking, Developing and Evaluating Chunks, Recursion in Linguistic Structure, Named Entity Recognition, Relation Extraction.

Text Books:

[1] Daniel Jurafsky, James H. Martin - Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, Pearson Publication, 2014.
[2] Steven Bird, Ewan Klein and Edward Loper - Natural Language Processing with Python, First Edition, OReilly Media, 2009.
[3] Christopher D.Manning and Hinrich Schuetze - Foundations of Statistical Natural Language Processing, MIT press, 1999.

Reference Books:

[1] Breck Baldwin, Language Processing with Java and LingPipe Cookbook, Atlantic Publisher, 2015.
[2] Richard M Reese, Natural Language Processing with Java, OReilly Media, 2015.
[3] Nitin Indurkhya and Fred J. Damerau, Handbook of Natural Language Processing, Second Edition, Chapman and Hall/CRC Press, 2010.
[4] Roland R.Hausser - Foundations of Computational Linguistics: Human Computer Communication in Natural Language, Paperback, MIT press,2011
[5] Tanveer Siddiqui, U.S. Tiwary, Natural Language Processing and Information Retrieval, Oxford University Press, 2008
[6] Daniel Jurafsky and James H. Martin - Speech and Language Processing, 2nd
[7] Edition , Prentice Hall,2008.
[8] Charu C.Aggarwal - Machine Learning for Text, Springer,2018 edition

6KS04 BIG DATA ANALYTICS [L-3, T-0, C-3]

Course Prerequisite: Knowledge of basic computer science principles and skills, Basic knowledge of Linear Algebra and Probability Theory, Basic knowledge of Data Base Management Systems

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Big Data Analytics by being able to do each of the following:

1. To know the fundamental concepts of big data and analytics.
2. To explore tools and practices for working with big data.
3. To know about the research that requires the integration of large amounts of data.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to

1. Work with big data tools and its analysis techniques.
2. Analyze data by utilizing clustering and classification algorithms.
3. Learn and apply different algorithms and recommendation systems for large volumes of data.
4. Perform analytics on data streams.
5. Learn NoSQL databases and management.

Unit I: Big Data Analytics and Lifecycle Hours: 6

Big Data Analytics: Big Data Overview, State of the Practice in Analytics, Key Roles for the New Big Data Ecosystem, Examples of Big Data Analytics, Data Analytics Lifecycle: Overview, Phase 1: Discovery, Phase 2: Data Preparation, Phase 3: Model Planning, Phase 4: Model Building, Phase 5: Communicate Results, Phase 6: Operationalize, Case Study: Global Innovation Network and Analysis (GINA).

Unit II: Review of Basic Data Analytics Methods, Clustering and Association Rules Hours: 7

Exploratory Data Analysis, Statistical Methods for Evaluation: Hypothesis Testing, Difference of Means, Wilcoxon Rank-Sum Test, Type I and II Errors, ANOVA, Overview of Clustering, K-means: Use Cases, Overview, Number of Clusters, Diagnostics, Additional Algorithms, Overview, Apriori Algorithm, Evaluation of Candidate Rules, Applications of Association Rules, An Example: Transactions in a Grocery Store, The Groceries Dataset, Frequent Itemset Generation, Rule Generation and Visualization, Validation and Testing, Diagnostics.

Unit III: Regression and Classification Hours: 7

Linear Regression: Use Cases, Model Description, Diagnostics, Logistic Regression: Use Cases, Model Description, Diagnostics, Reasons to Choose and Cautions, Additional Regression Models, Decision Trees: Overview of a Decision Tree, The General Algorithm, Decision Tree Algorithms, Evaluating a

Decision Tree, Decision Trees, Naïve Bayes: Bayes' Theorem, Naïve Bayes Classifier, Smoothing, Diagnostics, Naïve Bayes, Diagnostics of Classifiers, Additional Classification Methods.

Unit IV: Time Series Analysis and Text Analysis Hours: 6

Overview of Time Series Analysis: Box-Jenkins Methodology, ARIMA Model: Autocorrelation Function (ACF), Autoregressive Models, Moving Average Models, ARMA and ARIMA Models, Building and Evaluating an ARIMA Model, Reasons to Choose and Cautions, Additional Methods, Text Analysis Steps, A Text Analysis Example, Collecting Raw Text, Representing Text, Term Frequency—Inverse Document Frequency (TFIDF), Categorizing Documents by Topics, Determining Sentiments, Gaining Insights.

Unit V: Tool and Techniques: MapReduce & Hadoop Hours: 7

Big Data Tool and Techniques: Big Data Storage, High-Performance Architecture, HDFS, MapReduce and YARN, Big Data Application Ecosystem, Zookeeper, HBase, Hive, Pig, Mahout, Developing Big Data Applications: Parallelism, Myth, Application Development Framework, MapReduce Programming Model, Simple Example, More on MapReduce, Other Frameworks, The Execution Model, Analytics for Unstructured Data: Use Cases, MapReduce, Apache Hadoop, The Hadoop Ecosystem: Pig, Hive, HBase, Mahout, NoSQL.

Unit VI: Database Analytics, NoSQL and Graph Analytics Hours: 7

SQL Essentials, In-Database Text Analysis, Advanced SQL, NoSQL Data Management: What is NoSQL, Schemaless Models, Key-Value Stores, Document Stores, Tabular Stores, Object Data Stores, Graph Database, Communicating and Operationalizing an Analytics Project, Creating the Final Deliverables, Graph Analytics: Model, Triples, Graphs and Network Organization, Graph Analytics and Use Cases, Graph Analysis Algorithms, Technical Complexity, Features of Graph Analytic Platform, Data Visualization Basics.

Text Books:

[1] EMC Education Services, "Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data", 2015, John Wiley & Sons, Inc., ISBN: 978-1-118-87613-8.

[2] David Loshin, "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph", First Edition, 2013, Morgan Kaufmann/Elsevier Publishers, ISBN: 978-0-12-417319-4.

Reference Books:

[1] Bart Baesens, "Analytics in a Big Data World: The Essential Guide to Data Science and its Applications", First Edition, 2014, Wiley Publishers, ISBN: 978-1-118-89271-8.

[2] Mohammad Guller, "Big Data Analytics with Spark A Practitioner's Guide to Using Spark for Large-Scale Data Processing, Machine Learning, and Graph Analytics, and High-Velocity Data Stream Processing", First Edition, 2015, Apress Publisher, ISBN-13 (pbk): 978-1-4842-0965-3.

[3] Arshdeep Bahga & Vijay Madisetti, "Big Data Science & Analytics: A Hands-On Approach",
First Edition, 2019, ISBN: 978-1-949978-00-1.

6KS04 SENSORS AND ACTUATORS [L-3, T-0, C-3]

Course Prerequisite: Internet of Things, Micro-technology

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Sensors and Actuators by being able to do each of the following:

1. To understand the fundamentals of sensors and actuators
2. An exposure to sensors and its importance in the real world
3. To understand functional safety in machinery and emergency stop applications

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to

1. Fabricate some of those sensors
2. Simulate sensors and characterize before fabricating it
3. Design application with sensors and actuators for real world

Unit I: Hours: 7

Introduction: Sensors and Actuators, Technologies related to Sensors: Data Logger, Metal Detector, Photoelectric Sensor, Global Positioning System, Wireless Sensor Network, Sonar, Echo Sounding, Level Sensor, Biosensor, Blood Glucose Monitoring, Load Cell

Unit II: Hours: 7

Application of Sensors: On-board Automobile Sensors, Home Appliance Sensors, Aerospace Sensors, Sensors for Manufacturing, Medical Diagnostic Sensors, Sensors for Environmental Monitoring

Unit III: Hours: 7

Varied Types of Actuators: Pneumatic Actuator, Hydraulic Cylinder, Linear Actuator, Plasma Actuator, Rotary Actuator

Unit IV: Hours: 7

Actuators: Technologies and Devices- Pneumatic Motor, Pneumatic Cylinder, Hydraulic Press, Jackscrew, Hoist (Device), Electroactive Polymers, Roller Screw, MEMS Magnetic Actuator.

Unit V: Hours: 7

Remote Sensing: An Overview- Water Remote Sensing, Remote Sensing, Lidar, ERDAS Imagine, TerrSet, Remote Sensing (Archaeology)

Unit VI: Hours: 7

Rader and its application: Radar, Radar Imaging, Radar Navigation

Text Books:

[1] Princeton Brown, “Sensors and Actuators: Technology and Applications”, Library Press, 2017.

[2] D. Patranabis, “SENSORS AND TRANSDUCERS”, Second Edition, PHI Learning Private Limited, 2003.

Reference Books:

[1] D.A. Hall and C.E.Millar, “Sensors and Actuators”, CRC Press, 1999.

[2] Nathan Ida, “Sensors, Actuators, and their Interfaces: A multidisciplinary introduction (Materials, Circuits and Devices)”, Large Print, 2011.

6KSO4 Prof. Elect. II (iv) CRYPTOGRAPHY

Course Prerequisite: Discrete Structure & Graph Theory, Data Communication and Networking, Introduction to Cyber security

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Cryptography by being able to do each of the following:

1. Understand Security Concepts.
2. Know about various encryption techniques.
3. Understand the concept of public key cryptography.
4. Study message authentication and hash functions.
5. Impart knowledge on Network security, Internet Security Protocols.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to

1. Classify the symmetric encryption techniques
2. Illustrate various public key cryptographic techniques
3. Evaluate the authentication and hash algorithms.
4. Discuss authentication applications
5. Summarize the intrusion detection and its solutions to overcome the attacks.
6. Understand basic concepts of system level security

Unit I: (Hours7)

Cryptography: Concepts and Techniques Introduction, Plain Text and Cipher Text, Substitution and Transposition Techniques, Encryption and Decryption, Symmetric and Asymmetric Key Cryptography, Stenography, Key Range and Key Size, Possible Types of Attacks.

Unit II: (Hours7)

Symmetric Key Algorithms and AES: Introduction, Algorithm Types and Modes, An Overview of Symmetric Key Cryptography, Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), RC4, RC5, Advanced Encryption Standard(AES).

Unit III: (Hours7)

Asymmetric Key Algorithms, Digital Signatures and RSA: Introduction, History and Overview of Asymmetric Key Cryptography, The RSA Algorithm, Symmetric and Asymmetric Cryptography, Digital Signatures.

Unit IV: (Hours7)

Digital Certificates and Public Key Infrastructure (PKI): Introduction, Digital Certificates, Private Key Management, The PKIX Model, Public Key Cryptography Standards (PKCS), XML, PKI and Security, Creating Digital Certificate.

Unit V: (Hours7)

Internet Security Protocols: Introduction, Concepts, Secure Socket Layer (SSL), Transport Layer Security (TLS), Secure Hypertext Transport Protocol (SHTTP), Time Stamping Protocol (TSP), Secure Electronic Transaction (SET), SSL Versus SET, Electronic Money, Email Security.

Unit VI: (Hours7)

User Authentication: Introduction, Authentication Basics, Passwords, Authentication Tokens, Certificate-based- Authentication, Biometric Authentication.

Text Book: Atul Kahate, "Cryptography and Network Security", McGraw Hill, Second Edition.

Reference Books:

1. William Stallings, "Cryptography and Network Security, Principles and Practice", PHI Fourth Edition.
2. Behrouz A. Forouzan and Debdeep Mukhopadhyay,"Cryptography and Network Security", Mc-Graw Hill, Second Edition.
3. Matt Bishop, "Computer Security Arts and Science",Pearson Education.
4. Douglas R Stinson, "Cryptography, Theory and Practice" CRC Press.
5. Keith M Martin, "Everyday Cryptography, Fundamental Principles and Applications", Oxford University Press, Second Edition.

6KS06 DESIGN AND ANALYSIS OF ALGORITHMS – LAB [P-2, C-1]

Course Prerequisite: Any programming language, Discrete Mathematics and Data Structures

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Design and Analysis of Algorithms by being able to do each of the following:

1. To understand asymptotic analysis of algorithms.
2. To apply algorithmic strategies while solving problems.
3. Ability to analyze time and space complexity.
4. Demonstrate a familiarity with major algorithms.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to

1. Carry out the analysis of various Algorithms for mainly Time complexity.
2. Apply design principles and concepts to algorithm design.
3. Understand different algorithmic design strategies.
4. Analyze the efficiency of algorithms using time complexity.
5. Apply the standard sorting algorithms.

List of Experiments: This is the sample list of Experiments; minimum 12 experiments are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

List of Experiments based on Syllabus: (Maximum 20)

[1] Implement C programs to perform recursive calls using the following searching algorithms.

1. Linear Search when the list is given.
2. Binary Search when the given list is not sorted.

[2] Study and analyze to sort an array of integers using merge sort.

[3] Implement and analyze to sort an array of integers using quicksort.
[4] Write a program to implement the Closest Pair of Points problem using the divide and conquer strategy.
[5] Study and Implement the Divide and Conquer strategy using the Merge sort Algorithm and determine the complexity of an algorithm. DATA- {23, 12, 3, 5, 89, 1, 24}
[6] Write a C program for Implementing (n X n) matrix multiplication using the Strassen matrix multiplication algorithm.
[7] Explain the knapsack algorithm to find an optimal solution of getting maximum profit and implement using the program.
[8] Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm and implement using C.
[9] Implement programs to find minimum cost spanning trees from a given graph using Prim's algorithm.
[10]Implement Prim's algorithm to find the Minimum Cost Spanning Tree of an undirected graph using the program.
[11] Develop a program to implement Floyd's algorithm which will produce the shortest distance between all vertex pairs of a weighted graph.
[12] Implement programs to find the shortest path in a given graph using Dijkstra's algorithm.
[13] Implement programs factorial knapsack problem.
[14] Develop a program to implement Strassen's matrix multiplication algorithm.
[15] Implement programs to implement LCS problems using Dynamic Programming.
[16] Develop a program to implement matrix chain multiplication problems using dynamic programming.
[17] Explain Breadth-First Search and Implement BFS to print all the nodes reachable from a given starting node in a digraph.
[18] Develop a program to Print all the nodes reachable from a given starting node in a digraph using Depth First Search.
[19] Study an algorithm Tower of Hanoi where the aim is to move the entire stack to another rod for n=3 and understand the concept of recursion.
[20] Implement C programs N Queen's problem using Back Tracking.

List of Experiments beyond Syllabus: (Maximum 05)
[1] Implement the Work Function Algorithm and the Greedy Algorithm for the k-Server problem on graph metrics.
[2] Design and Implement Boyer Moore Algorithm for Pattern Searching.
[3] Design and Implement Topological Sort of a graph using departure time of vertex.
[4] Implement programs to find an s-t cut of minimum capacity. Minimum Cut Problem s 2 3 4 5 6 7 t 15 5 30 15 10 8 15 9 6 10 15 4 4 A Capacity = 10 + 8 + 10 = 28
[5] Implement programs to s-t flow of maximum value. Maximum Flow Problem 10 9 9 14 4 10 4 8 9 1 0 0 0 14 capacity flow s 2 3 4 5 6 7 t 15 5 30 15 10 8 15 9 6 10 15 4 4 0 Value = 28
Text Books:
[1] Dave and Dave: "Design and Analysis of Algorithms" Pearson Education.
Reference Books:
[1] Aho,Hopcroft & Ullman "The Design & Analysis of Computer Algorithms", Addison-Wesley
[2] G. Brassard, P.Bratley: "Fundamentals of Algorithmics" , PHI
[3] Horowitz & Sahani: " Fundamental Algorithms" , Galgotia.
[4] Cormen, T.H, Lierson & Rivest: " Introduction to Algorithms", Mc Graw-Hill.

6KS07 SOFTWARE ENGINEERING LAB.
Course Prerequisite: A Scripting Language, IDEs (Integrated Development Environment), Databases, Software Development Life Cycle (SDLC)
Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Software Engineering by being able to do each of the following:

1. Impart state-of-the-art knowledge on Software Engineering and UML in an interactive manner
2. Present case studies to demonstrate the practical applications of different concepts
3. Provide a scope to the students where they can solve small, real-life problems
4. All the while it is intended to present Software Engineering as an interesting subject to the students where learning and fun can go alongside.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to
1. Understand basic Software engineering methods and practices, and their appropriate application.
2. Describe software process models such as the waterfall and evolutionary models.
3. Discuss role of project management including planning, scheduling and, risk management.
4. Explain data models, object models, context models and behavioral models.
5. Understand of different software architectural styles and Process frame work.

List of experiments: This is the sample list of Experiments; minimum 12 experiments are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

[1] Identifying the Requirements from Problem Statements

Requirements, Characteristics of Requirements, Categorization of Requirements, Functional Requirements, Identifying Functional Requirements

[2] Estimation of Project Metrics

Project Estimation Techniques, COCOMO, Basic COCOMO Model, Intermediate COCOMO Model, Complete COCOMO Model, Advantages of COCOMO, Drawbacks of COCOMO, Halstead's Complexity Metrics

[3] Modeling UML Use Case Diagrams and Capturing Use Case Scenarios

Use case diagrams |,Actor , Use Case , Subject , Graphical Representation , Association between Actors and Use Cases , Use Case Relationships , Include Relationship , Extend Relationship , Generalization Relationship ,Identifying Actors , Identifying Use cases , Guidelines for drawing Use Case diagrams

[4] E-R Modeling from the Problem Statements

Entity Relationship Model , Entity Set and Relationship Set , Attributes of Entity , Keys , Weak Entity , Entity Generalization and Specialization ,Mapping Cardinalities , ER Diagram , Graphical Notations for ER Diagram , Importance of ER modeling

[5] Identifying Domain Classes from the Problem Statements

Domain Class , Traditional Techniques for Identification of Classes ,Grammatical Approach Using Nouns ,Advantages , Disadvantages ,Using Generalization ,Using Subclasses , Steps to Identify Domain Classes from Problem Statement , Advanced Concepts

[6] State chart and Activity Modeling

State chart Diagrams , Building Blocks of a State chart Diagram , State , Transition , Action , Guidelines for drawing State chart Diagrams , Activity Diagrams , Components of an Activity Diagram, Activity , Flow , Decision , Merge , Fork ,Join , Note , Partition ,A Simple Example , Guidelines for drawing an Activity Diagram

[7] Modeling UML Class Diagrams and Sequence diagrams

Structural and Behavioral aspects , Class diagram , Elements in class diagram , Class , Relationships , Sequence diagram , Elements in sequence diagram , Object , Life-line bar , Messages

[8] Modeling Data Flow Diagrams

Data Flow Diagram, Graphical notations for Data Flow Diagram, Explanation of Symbols used in DFD , Context diagram and leveling DFD

[9] Estimation of Test Coverage Metrics and Structural Complexity

Control Flow Graph, Terminologies, McCabe's Cyclomatic Complexity, Computing Cyclomatic Complexity, Optimum Value of Cyclomatic Complexity, Merits , Demerits

[10] Designing Test Suites

Software Testing , Standards for Software Test Documentation , Testing Frameworks , Need for Software Testing , Test Cases and Test Suite , Types of Software Testing , Unit Testing , Integration Testing , System Testing , Example , Some Remarks.

Software Requirements: StarUML

Text Book: Pressman Roger. S: Software Engineering, A Practitioner's Approach, TMH.

Reference Books:

[1] Somerville: Software Engineering (Addison-Wesley) (5/e)
[2] Fairly R: Software Engineering (McGraw Hill)
[3] Davis A: Principles of Software Development (McGraw Hill)
[4] Shooman, M.L: Software Engineering (McGraw-Hill).

6KS09 C SKILL LAB IV– LAB (DevOps) [P-2, C-1]

Course Prerequisite: Basic knowledge on SDLC and STLC

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of DevOps learning by being able to do each of the following:

1. Learn what Jenkins, continuous integration is and where does Jenkins fits into SDLC (Software Development Life Cycle)
2. Learn how to setup Jenkins and use Jenkins on their systems, create and configure jobs in Jenkins
3. Learn how to use and manage plugins, how to create and manage users in Jenkins
4. Learn how to deploy application on server, how to work with multiple nodes
5. Learn how to create pipelines

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to

1. Install and setup of Jenkins on your systems
2. Create and run jobs in Jenkins
3. Add and manage plugins. Use plugins in jobs
4. Create and run pipelines in Jenkins
5. Setup, configure, and deploy jobs

List of Experiments: This is the sample list of Experiments; minimum 12 experiments are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)

List of Experiments based on Syllabus: (Maximum 20)

1. Study and implement Linux commands
2. Study practical on installation of java, Tomcat Server
3. Study practical on software development life cycle
4. Study practical on DevOps life cycle & stages
5. Study practical on DevOps Tools (Docker, Jenkins, Git, Jira, copado)
6. Learn about DevOps Pipeline (CI /CD) using any tool
7. Study Practical on AWS for DevOps
8. Study Practical on Microsoft Azur for DevOps
9. Study Practical on Google Cloud for DevOps
10. Study Practical on Salesforce with Copado for DevOps
11. To setup and configure of Jenkins
12. To create Job and manage it using Jenkins
13. To experiment plugin management with jenkins
14. To study and demonstrate User role creation and management using Jenkins
15. To study and demonstrate Integration with Git using Jenkins
16. To study and demonstrate Automated deployments using Jenkins
17. To study and demonstrate Build and delivery pipelines using Jenkins
18. To study and demonstrate Job Parameterization using Jenkins
19. To study and demonstrate Command line executions using Jenkins
20. To study and demonstrate Jenkins node management

List of Experiments beyond Syllabus: (Maximum 05)
1. Learn how to setup Jenkins on docker
2. Learn how to do Jenkins maintenance
3. Learn how to work with Git and Jenkins

Text Book: John Ferguson Smart: Jenkins: The Definitive Guide, O'Reilly Media, Inc.

Reference Books:

[1] Gene Kim, Jez Humble, Patrick Debois, and John Willis,: The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations

[2] Gene Kim, Kevin Behr, and George Spafford,: The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win,

[3] Andrew Davis, : Mastering Salesforce DevOps: A Practical Guide to Building Trust While Delivering Innovation, Apress

6KS08 EMERGING TECHNOLOGY LAB II

6KS08 Emerging Technology Lab II is based on 6KS04 Professional Elective-II. Tentative FOSS Tools & Technology for Practical’s are as follows:

AI : Natural Language Toolkit (NLTK),SpaCy, PyTorch-NLP, Natural, Retext, TextBlob
DS : KNIME, Spark, Neo4J, MongoDB, Hive, Storm,
IoT : Devicehub, Zetta, Node-RED, Flutter, M2MLabs Mainspring
Cyber Security : VeraCrypt, ModSecurity, AdBlocker, CheckShortURL, SPAMfighter, SpamBully

B.E. Sem. VII (Computer Science & Engineering)

7KS01 / 7KE01 SOCIAL SCIENCES AND ENGINEERING ECONOMICS

Course Objectives:

The phenomenal progress of technology in the twentieth century has brought dramatic changes in human lifestyles from the social and economic point of view. This subject helps students to get an understanding of market trends, economic transformations, changes in the laws and equip them to have a better understanding of the market.

Course objectives are:

1. To help students to understand the importance of economics to engineers
2. To let them know about the Indian Parliament
3. To enhance their knowledge about culture and civilization
4. To help students to get an understanding of Market Trends, Economic Transformations, Changes in the Laws & equip them to have a better understanding of Market
5. To critically examine the market trends.

Course Outcomes:

At the end of the course, students will have-

1. An ability to understand the importance of social science and economics in professional life.
2. An ability to utilize high-level interpersonal skills to negotiate with stakeholders and maintain cordial relationships with them reflecting the professional ethics and responsibilities.
3. Understanding of professional responsibility with socioeconomic constraints and norms
4. An ability to understand the need of society and design the system to fulfil it with deep analysis.
5. An ability to understand the social science and engage in a lifelong learning process performing better in the group as well as individually.

SECTION - A

Unit I : Study of Social Science : Importance to Engineer, salient features of Indian constitution. Fundamental Rights and Duties. Directive Principles of State Policy. (8)

Unit II : Indian Parliament : Composition and powers, President of India : Election and Powers. Council of Ministers and Prime Minister (8)

Unit III : Impact of Science and Technology on culture and Civilization. Human Society: Community Groups. Marriage and Family: Functions, Types and problems. (8)

SECTION – B

Unit IV: Production : Factors of production, Laws of return, Forms of Business Organization. (8)

Unit V: Banking : Functions of Central and Commercial Banks. Introduction to GST, Market : Forms, perfect, imperfect competition and monopoly. (8)

Unit VI: Nature and scope of Economics : Special significance of Economics to Engineers. Economics of Development : Meaning, Characteristics of under development, obstacles to Economic growth and vicious circle of poverty. (8)

Books Recommended :

1. Pylee M.V. : Constitutional Govt. in India, S.Chand and Co.
2. C N Shankar Rao: Sociology, S.Chand and Co.
3. Dewett and Varma J.D. : Elementary Economic Theory, S.Chand and Co.
4. A.N.Agrawal : Indian Economy, Problem of Development and Planning (Wiley Eastern Ltd), New Delhi.
5. S.K.Mishra : Indian Economy, Its Development Experience. Himalaya Pub.House, Bombay.
6. E.Kuper : Economics of W.R. Development, McGraw Hill Co.,
7. Brij Kishore Sharma. : The Constitution of India, PHI.
8. Mahajan : The Constitution of India, S.Chand, New Delhi.
9. Maclaver and Page : Principle of Sociology.
10. Davis K. : Human Society
11. Datt R.K. : Indian Economy, S.Chand and Comp. New Delhi P.M.Sundharam
12. Dhingra I.C. : Indian Economy
13. Jemes L.E., R.R.Lee : Economics of W.R.Planning, McGraw Hill Co.

7KS02 COMPUTER GRAPHICS

Course Prerequisite: Data Structures and algorithms, Basic Mathematics, Geometry, linear algebra, vectors and matrices.

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of Computer Graphics by being able to do each of the following:

- To acquaint the learner with the basic concepts of Computer Graphics.
- To learn the various algorithms for generating and rendering graphical figures.
- To get familiar with mathematics behind the graphical transformations.
- To understand various methods and techniques regarding projections, animation, shading, illumination, and lighting

Course Outcomes (Expected Outcome):

On completion of the course, the students will be able to:

1. Describe the basic concepts of Computer Graphics.
2. Demonstrate various algorithms for basic graphics primitives.
3. Apply 2-D geometric transformations on graphical objects.
4. Use various Clipping algorithms on graphical objects.
5. Explore 3-D geometric transformations, curve representation techniques and projections methods.
6. Explain visible surface detection techniques and Animation.

Unit I: Introduction and Overview of Graphics System: (Hours:7)

Definition and Representative uses of computer graphics, Overview of coordinate system, Definition of scan conversion, rasterization and rendering. Raster scan & random scan displays, Architecture of raster graphics system with display processor, Architecture of random scan systems.

Unit II: Output Primitives: (Hours:7)

Scan conversions of point, line, circle: DDA algorithm and Bresenham algorithm for line drawing, midpoint algorithm for circle; Aliasing, Antialiasing techniques like Pre and post filtering, super sampling, and pixel phasing.

Unit III: Two Dimensional Geometric Transformations: (Hours:7)

Basic transformations: Translation, Scaling, Rotation , Matrix representation and Homogeneous Coordinates Composite transformation Other transformations: Reflection and Shear

Unit IV: Two-Dimensional Viewing and Clipping: (Hours:7)

Viewing transformation pipeline and Window to Viewport coordinate transformation , Clipping operations: Point clipping, Line clipping algorithms: Cohen-Sutherland, Liang: Barsky.

Unit V: Three Dimensional Geometric Transformations, Curves and Fractal Generation: (Hours:7)

3D Transformations: Translation, Rotation, Scaling and Reflection, Composite transformations: Rotation about an arbitrary axis, Projections – Parallel, Perspective. (Matrix Representation), Bezier Curve, B-Spline Curve.

Unit VI: Visible Surface Detection and Animation: (Hours:7)

Visible Surface Detection: Classification of Visible Surface Detection algorithm, Back Surface detection method, Depth Buffer method, Area Subdivision method Animation: Introduction, Design of animation sequences, Animation languages, Keyframe, Morphing, Motion specification.

Text Book: Hearn, Baker, "Computer Graphics (C version)" – Pearson Education.

Reference Books:

1. J. Foley, V. Dam, S. Feiner, J. Hughes, Computer Graphics Principles and Practice‖, 2nd Edit ion, Pearson Education, 2003, ISBN 81 – 7808 – 038 – 9.
2. D. Rogers, J. Adams, Mathematical Elements for Computer Graphics‖, 2nd Edition, T ata Mc-Graw Hill. Publication, 2002, ISBN 0 – 07 – 048677 – 8.
3. Mario Zechner, Robert Green, Beginning Android 4 Games Development‖, Apress, ISBN: 97 8-81-322- 0575- 3.

7KS03 CLOUD COMPUTING (L-4, T-0, C-4)

Course Prerequisite: Data Communication and Networks

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Cloud Computing by being able to do each of the following:

· To provide students with the fundamentals and essentials of Cloud Computing.

· To provide students a foundation of Cloud Computing, Cloud Computing services and tools in real life scenarios.

· To enable student to explore some important Cloud Computing driven commercial systems and applications.

· To provide students with essentials of Cloud Computing architecture, Virtualization, Storage and Network concepts.

Course Outcomes (Expected Outcomes): On completion of the course, the students will be able to:

1. Describe the fundamental concept, architecture and applications of Cloud Computing.
2. Discuss the problems related to cloud deployment model.
3. Examine the concept of virtualization.
4. Identify the role of network connectivity in the cloud.
5. Assess different Cloud service providers.
6. Inspect the security issues in cloud service models.

Unit I: Cloud Computing Fundamental, Architecture and Management: Hours: 8

Computing Paradigm and various computing types, Cloud Computing Fundamentals: Motivation for Cloud Computing, The need for Cloud Computing, Defining Cloud Computing, Principles of Cloud Computing, Requirements of Cloud Services, Cloud Applications, Benefits and Drawbacks. Cloud Computing Architecture and

Management: Introduction, Cloud Architecture, Network connectivity in Cloud Computing, Applications on the cloud, Managing Cloud, Migrating Application to cloud.

Unit II: Cloud Deployment and Service Models: Hours: 8

Cloud Deployment Models: Introduction, Private Cloud, Public Cloud, Community Cloud, Hybrid Cloud. Cloud Service Models: Introduction, Infrastructure as a Service, Platform as a Service, Software as a Service, Other Cloud Service Models.

Unit III: Operating System and Virtualization: Hours: 8

Types of Operating Systems, Role of OS in Cloud Computing, Features of Cloud OS. Application Environment: Need for Effective ADE, Application Development Methodologies, Cloud Application

Development Platforms and Cloud Computing API's. Virtualization: Introduction, Virtualization Opportunities, Approaches to Virtualization, Hypervisors, Virtualization to Cloud Computing.

Unit IV: Software Development in Cloud and Networking for Cloud Computing: Hours: 8

Introduction, Different Perspectives on SaaS Development, New Challenges, Cloud-Aware Software Development Using PaaS Technology. Networking for Cloud Computing: Introduction, Overview of Data Center Environment, Networking Issues in Data Centers, Transport Layer Issues in DCNs.

Unit V: Cloud Service Providers: Hours: 8

Introduction, EMC: IT, and captive cloud toolkit, Google: Platform, Storage, Cloud connects, Cloud Print and App Engine, Amazon Web Services: Elastic Compute Cloud, Simple storage, Simple Queue Service, Microsoft: Windows Azure, IBM Cloud models and IBM Smart Cloud, SAP Labs: SAP HANA Cloud Platform, Virtualization Services Salesforce: Sales Cloud and Service Cloud, Rackspace and VMware.

Unit VI: Open-Source Support for Cloud and Security in Cloud Computing : Hours: 8

Open-Source Support for Cloud: Introduction, Open Source Tools for IaaS, Open Source Tools for PaaS, Open Source Tools for SaaS, Open Source Tools for Research, Distributed Computing Tools for Management of Distributed Systems. Security in Cloud Computing: Introduction, Security Aspects: Data, Virtualization and Network Security, Platform-Related Security: Security issues in Cloud Service Models, SaaS, PaaS, IaaS security issues, Audit and Compliance: Disaster Recovery, Privacy and Integrity.

Text Book: K. Chandrasekaran: Essentials of Cloud Computing, Edition, CRC Press Taylor & Francis Group.

Reference Books:

1. A. Shrinivasan, J. Suresh: Cloud computing a practical approach for learning and implementation, Pearson publication.
2. M. N. Rao: Cloud Computing, PHI Learning Pvt. Ltd, 2015.
3. Dr. Kumar Saurabh: Cloud computing, 2nd Edition, Wiley India 2012.
4. Rajkumar Buyya, James Broberg and Andrzej M. Goscinski: Cloud Computing: Principles and Paradigms, John Wiley & Sons, Inc. 2011.
5. Anthony T. Velte , Toby J. Velte and Robert Elsenpeter, Cloud computing a practical approach, Tata McGraw- Hill , New Delhi – 2010.
6. Judith Hurwitz, Robin Bloor, Marcia Kaufman and Fern Halper, “Cloud computing for dummies” Wiley Publishing, Inc, 2010.

7KS04 ROBOTICS (L-3, T-0, C-3)

Course Prerequisite: Mathematics

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Robotics by being able to do each of the following:

· To introduce the functional elements of Robotics
· To impart knowledge on the direct and inverse kinematics
· To introduce the manipulator differential motion and control
· To educate on various path planning techniques
· To introduce the dynamics and control of manipulators

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to

1. Describe basic concept of robotics.
2. Explain Components of a Robot System & Mechanical Systems
3. Illustrate Control of Actuators in Robotic Mechanisms
4. Compare and contrast Robotic Sensory Devices
5. Recommend Robotics Hardware & Software Considerations in Computer Vision
6. Design Robotic system by taking real time considerations.

Unit I: Introduction to Robotics: Objectives, Motivation, Historical Perspective of Robots, Classification of Robots, Classification by Control Method, Continuous-path servo-controlled robots, Major Components of a Robot, Fixed versus Flexible Automation. (**Hours: 7)**

Unit II: Components of a Robot System & Mechanical Systems: Basic Components of a Robot System, Functions of a Robot System Specifications of Robot Systems, Kinematic Chains the Manipulator End Effectors, Resolution, Forces Encountered in Moving Coordinate Systems Lagrangian Analysis of a Manipulator. (**Hours: 7)**

Unit III: Control of Actuators in Robotic Mechanisms: Closed-Loop Control in a Position Servo, the Effect of Friction and Gravity, Frequency-Domain Considerations, Control of a Robotic Joint Brushless DC Motors, Direct- Drive Actuator, Hydraulic Actuators. (**Hours: 7)**

Unit IV: Robotic Sensory Devices: Non-Optical-Position Sensors, Optical Position Sensors, Robot Calibration Using an Optical Incremental Encoder, Instability Resulting from Using an Incremental Encoder, Velocity Sensors, Accelerometers. **(Hours: 7)**

Unit V: Computer Vision for Robotics Systems: A Functional Approach: Imaging Components, Image Representation, Hardware Considerations, Picture Coding, Object Recognition and Categorization, Software Considerations, Need for Vision Training and Adaptations. **(Hours: 7)**

Unit VI: Computer Considerations for Robotic Systems: Architectural Considerations, Hardware Considerations, Computational Elements in Robotic Applications Real-Time Considerations, Robot Programming, Path Planning, The Robot's Computer System. **(Hours: 7)**

Text Books:

1. Richard D.Klafter Thomas , Achmielewski and Michael Negin Robotic Engineering- An Integrated Approach Prentice Hall India – New Delhi.
2. Saeed B Nikku Introduction to Robotics , analysis control and applications Wiley-India 2nd Edition-2011

Reference Books:

1. B.K.Ghosh, Control in Robotics and Automation: Sensor Based Integration, Allied Publishers, Chennai, 1998.
2. S.Ghoshal, " Embedded Systems & Robotics" – Projects using the 8051 Microcontroller", Cengage Learning, 2009.
3. David Jefferis, "Artificial Intelligence: Robotics and Machine Evolution", Crabtree Publishing Company, 1992.
4. Robin Murphy, Robin R. Murphy, Ronald C. Arkin, "Introduction to AI Robotics", MIT Press, 2000.
5. Francis.X.Govers, "Artificial Intelligence for Robotics", Packt Publishing, 2018.
6. Huimin Lu, Xing Lu, "Artificial Intelligence and Robotics", Springer, 2017.
7. Lentin Joseph, "Robot Operating Systems (ROS) for Absolute Beginners, Apress, 2018
8. Aaron Martinez, Enrique Fernández, "Learning ROS for Robotics Programming", Packt Publishing Ltd, 2013.
9. Wyatt Newman, "A Systematic Approach to learning Robot Programming with ROS", CRC Press, 2017.
10. Ashitava Ghoshal, Robotics-Fundamental Concepts and Analysis', Oxford University Press, Sixth impression, 2010.
11. K. K.AppuKuttan, Robotics, I K International, 2007.
12. Edwin Wise, Applied Robotics, Cengage Learning, 2003.
13. Richard D. Klafter, Thomas .A, ChriElewski, Michael Negin - Robotic Engineering–An Integrated Approach, Prentice Hall of India, New Delhi, 2009.
14. S.R. Deb, Robotics Technology and flexible automation, Tata McGraw-Hill Education., 2009
15. Mikell P Groover & Nicholas G Odrey, Mitchel Weiss, Roger N Nagel, Ashish Dutta, Industrial Robotics, Technology programming and Applications, McGraw Hill, 2012
16. Deb. S. R. "Robotics technology and flexible automation", Tata McGraw Hill publishing company.

7KS04 Prof. Elect. III: (ii) DATA WAREHOUSE AND MINING

Course Prerequisite: Basic knowledge of Database management system

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of Data Warehouse and Mining by being able to do each of the following:

- Introduce the basics of data mining, data types, similarity and dissimilarity measures.
- Explain association rules and algorithms.
- Be familiar with mathematical foundations of data mining tools.
- To identify the scope and essentiality of Data Warehousing and Mining
- Demonstrate the appropriate data mining techniques for decision making.
- To develop research interest towards advances in data mining.

Course Outcomes (Expected Outcome):

On completion of the course, the students will be able to

1. Explain the basics of data mining techniques.
2. Identify the similarity and dissimilarity between the data sets.
3. Apply Data Preprocessing techniques.
4. Describe Data Warehouse fundamentals, Data Mining Principles.
5. Illustrate Multidimensional Data Analysis in Cube Space
6. Assess Mining Frequent Patterns, Associations, and Correlations

Unit I: Introduction: (Hours:7)

Why Data Mining? What Is Data Mining? , What Kinds of Data Can Be Mined? What Kinds of Patterns Can Be Mined? Which Technologies Are Used? , Which Kinds of Applications Are Targeted? , Major Issues in Data Mining.

Unit II: Getting to Know Your Data: (Hours:7)

Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity.

Unit III: Data Preprocessing: (Hours:7)

Data Preprocessing: An Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization.

Unit IV: Data Warehousing and Online Analytical Processing: (Hours:7)

Data Warehousing and Online Analytical Processing: Data Warehouse: Basic Concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, data lake and Data Ocean.

Unit V: Data Cube Technology: (Hours:7)

Data Cube Computation: Preliminary Concepts, Data Cube Computation Methods, Processing Advanced Kinds of Queries by Exploring Cube Technology, Multidimensional Data Analysis in Cube Space.

Unit VI: Mining Frequent Patterns, Associations, and Correlations: (Hours:7)

Basic Concepts and Methods: Basic Concepts, Frequent Itemset Mining Methods , Which Patterns Are Interesting?- Pattern Evaluation Methods .

Text Book:

Data Mining – Concepts and Techniques, Jiawei Han & Micheline Kamber, Morgan Kaufmann (MK)
Publishers, Elsevier, 3rd Edition, 2006.

Reference Books:

1. Data Mining Techniques, Arun K Pujari, 3rd edition, Orient Blackswan/Universities Press, 2013.
2. Data Warehousing Fundamentals, Paulraj Ponnaiah, John Wiley & Sons, 2001.
3. Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Pearson Education, 2007
4. Insight into Data mining Theory and Practice, K.P. Soman, Shyam Diwakar and V. Ajay, Easter Economy Edition, Prentice Hall of India, 2006.
5. G. K. Gupta, "Introduction to Data Mining with Case Studies", Easter Economy Edition, Prentice Hall of India, 2006.

7KS04 EMBEDDED SYSTEM (L-3, T-0, C-3)

Course Pre-requisite: Microprocessor and Assembly Language Programming, Computer Architecture and Organization

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Embedded System by being able to do each of the following:

1. Introduce the fundamentals and building blocks of Embedded System.
2. Impart the knowledge of basic embedded programming in various languages as well as data structures.
3. Introduce hardware units, bus communication in processors and input/output interfacing.
4. Impart knowledge of real-time operating system and various task scheduling algorithms.
5. Introduce basics of real-time operating system and case study example to elaborate importance of real-time operating system.

Course Outcomes (Expected Outcomes): On completion of the course, the students will be able to:

1. Describe the basics of embedded systems and structural core units as well as memory organization for embedded system.
2. Explain components of embedded system, characteristics and quality attributes of embedded systems.
3. Discuss role of 8051 microcontroller and its architecture in design of embedded systems
4. Examine the different Addressing modes and Instruction Set of 8051 microcontrollers.
5. Use knowledge of C programming to do embedded programming.
6. Assess the Real-Time Operating System concepts with VxWorks RTOS.

UNIT I: Introduction to Embedded System: What is Embedded System, Embedded Systems Vs General Computing Systems, History, classification, major application areas and purpose of Embedded Systems, Wearable Devices. The Typical Embedded System: Core of the Embedded System, Memory. **(Hours: 7)**

UNIT II: The Typical Embedded System: Sensors & Actuators, Communication Interface, Embedded Firmware, Other System Components, PCB and Passive Components. Characteristics of an Embedded System, Quality Attributes of Embedded Systems. Embedded Systems Application and Domain Specific Examples: Washing machine, Automotive. **(Hours: 7)**

UNIT III: Designing Embedded Systems with 8-bit Microcontroller - 8051: Factors to be considered in Selecting a Controller. Why 8051 Microcontroller. Designing with 8051 Microcontroller: 8051 Architecture, 8051 Memory Organization, Registers, Oscillator Unit, Ports, 8051 Interrupt System, Timer units, the Serial Port, 8051 Power Saving Modes. **(Hours: 7)**

.

UNIT IV: Programming the 8051 Microcontroller: Different Addressing modes supported by 8051. The 8051 Instruction Set: Data transfer instructions, Arithmetic instructions, Logical instructions, Boolean instructions, and Program Control Transfer instructions. Embedded Firmware Design Approaches, Assembly Language based Embedded Firmware development. **(Hours: 7)**

UNIT V:

Programming in Embedded C: Review of various constructs in C. Constant declarations, 'volatile' type qualifier, Delay generation and Infinite loops in Embedded C. Coding Interrupt Service Routines, Recursive and Re-entrant Functions, Dynamic memory allocation. **(Hours: 7)**

UNIT VI:

VxWorks Real Time Operating System (RTOS): How to choose an RTOS, Characteristics, Real Time Kernel, Hard/Soft Real time. VxWorks Task Creation, Management and Task Scheduling, Kernel Services, Inter Task Communication, VxWorks Task Synchronization and Mutual Exclusion, Interrupt Handling, Watchdog for task Execution monitoring, Timing and Reference in VxWorks.

The Embedded Product Development Life Cycle (EDLC): What is EDLC, Why EDLC, Objectives of EDLC, Different Phases of EDLC, EDLC approaches. **(Hours: 7)**

Text Book: Shibu K V "Introduction to Embedded Systems", Second Edition, McGraw-Hill.

References:

1. Rajkamal, "Embedded Systems, Architecture, Programming & Design", Third Edition, TMH.
2. Tammy Noergaard, "Embedded Systems Architecture" Elsevier Newness Publication.
3. Vahid and Givargis, "Embedded System Design" John Wiley & Sons P Ltd.

4. Peter Marwedel, “Embedded Systems Design” Springer, Netherland.
5. Jane W. S. Liu, “Real Time Systems”, Pearson Education.
6. Mohammad Ali Mazidi, “The 8051 Microcontroller and Embedded System using Assembly and C” Pearson

7KS04 PE III: DIGITAL FORENSICS

Course Prerequisite: Data Communication & Networking, Introduction to Cyber Security, Cryptography

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of Digital Forensics by being able to do each of the following:

- To understand the basic digital forensics and techniques for conducting forensic examinations on different digital
devices.
- To understand how to examine digital evidence such as data acquisition and identification analysis.
- To understand the basics of mobile phone forensics.
- To understand network based cyber security intrusion detection.
- To know the various forensics tools.

Course Outcomes (Expected Outcome):

On completion of the course, the students will be able to:

1. Describe Digital Forensics and its related preparation.
2. Outline Data Acquisition tools
3. Use knowledge to improve crime investigations.
4. Examine Digital Forensic and its validation.
5. Assess role of email and social media in investigations
6. Discuss Cloud Forensics.

Unit I: (Hours7)

Introduction: An Overview of Digital Forensics, Preparing for Digital Investigations, Preparing A Digital Forensics Investigations, Procedure for Private Sector High-Tech investigations, understanding data recovery workstation and software, conducting and investigations.

Unit II: (Hours7)

Data Acquisition: Understanding storage formats for digital evidence, determining the best acquisition method, Contingency planning for Image acquisition, acquisition tools, validating data acquisitions, performing RAID data acquisitions, remote network acquisition tools, other forensics acquisitions tools.

Unit III: (Hours7)

Processing Crime and Incident Scenes: Identifying Digital Evidence, Collection Evidence in Private Sector Scenes, Processing Law Enforcement Crime Scenes, Preparing for a search, Securing a Digital Incident or Crime Scene, Seizing Digital Evidence at the scene, Storing a Digital Evidence, Obtaining a Digital Hash,

Unit IV: (Hours7)

Digital Forensic Analysis and Validation: Data to collect and analyze, Validating Forensic data, Addressing data hiding techniques, Virtual Machine Forensics, Live Acquisition and Network Forensics

Unit V: (Hours7)

Email and Social Media Investigations: Role of Email in investigations, Roles of Client and server in Email, Investigating Emails Crimes and Violations, Email Servers, Specialize Email Forensic Tools, Digital Forensics to Social Media Communications.

Unit VI: (Hours7)

Cloud Forensics: Cloud Computing, Legal Challenges in Cloud Forensics, Technical Challenges in Cloud Forensics, Acquisitions in the cloud, conducting a cloud investigation, Tools for Cloud Forensics.

Text Book:

Nelson, B, Phillips, A, Stuart, C., “Guide to Computer Forensics and Investigations”, 6th Ed., Cengage Learning.

Reference Books:

1. Warren G. Kruse II and Jay G. Heiser, “Computer Forensics: Incident Response Essentials”, Addison Wesley, 2002.
2. Davidoff, S. and Ham, J., Network Forensics Tracking Hackers through Cyberspace, Prentice Hall, 2012.
3. Michael G. Solomon, K Rudolph, Ed Tittel, Broom N., and Barrett D., Computer Forensics Jump Start, Willey Publishing, Inc., 2011.
4. Marcella, Albert J., Cyber forensics: A field manual for collecting, examining and preserving evidence of computer crimes, New York, Auerbach publications, 2008.
5. Davidoff, Sherri, Network forensics: Tracking hackers through cyberspace, Pearson education India private limited, 2017.
6. John Sammons, The Basics of Digital Forensics, Elsevier, 1st Edition, 2015.

7KS05 (Prof. Elect. - IV (i)) BLOCKCHAIN FUNDAMENTALS (L-3, T-0, C-3**)**

Course Pre-requisite:Basic Knowledge of Distributed systems and Networking, Basic knowledge of Data Structure

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of blockchain fundamental by being able to do each of the following:

1. A comprehensive understanding of how blockchain systems (mainly Bitcoin and Ethereum) work,
2. To securely interact with them
3. Design, build, and deploy smart contracts and distributed applications,
4. Integrate ideas from blockchain technology into applications.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:

1. Understand the concept of decentralization of the block chain with different layers of blockchain
2. Apply basic cryptographic primitives with encryption standards.
3. Analyze & Design Consensus Algorithms.
4. Examine fundamentals of Bitcoin, how Bitcoin transactions are constructed and used with Bitcoin addresses, accounts, and mining
5. Understand foundation, architecture, and use of the Ethereum blockchain.
6. Execute & build block chain application/ transaction

Unit I: Blockchain Fundamentals (Hours: 7)

Introduction to Blockchain, History, Centralised versus Decentralised systems, Layers of blockchain, Importance of blockchain, Blockchain uses and use cases.

Unit II:Blockchain Working with Cryptography (Hours: 7)

Laying the Blockchain Foundation, Cryptography, Symmetric Key Cryptography, DES cryptography, Advanced Encryption Standard, Cryptographic Hash Functions, MAC and HMAC, Asymmetric Key Cryptography, Diffie- Hellman Key Exchange, Symmetric vs. Asymmetric Key Cryptography

Unit III:Consensus Algorithms (Hours: 6)

Introducing the consensus problem, Analysis and design, Classification, Algorithms: CFT algorithms, BFT algorithms, Choosing an algorithm

Unit IV:Bitcoin & Its Working (Hours: 7)

The History of Money, Dawn of Bitcoin: What Is Bitcoin, Working with Bitcoins. The Bitcoin Blockchain: Block Structure, The Genesis Block. The Bitcoin Network: Network Discovery for a New Node Bitcoin Transactions, Bitcoin Wallets

Unit V: Ethereum (Hours: 7)

From Bitcoin to Ethereum, Ethereum as a Next-Gen Blockchain Design Philosophy of Ethereum Enter the Ethereum Blockchain Ethereum Blockchain Ethereum Accounts Trie Usage Merkle Patricia Tree RLP Encoding Ethereum Transaction and Message Structure. Ethereum Smart Contracts Contract Creation

Unit VI: Blockchain Application Development (Hours: 6)

Decentralized Applications, Blockchain Application Development, interacting with the Bitcoin Blockchain, Interacting Programmatically with Ethereum, Sending Transactions

Text Books:

1. Beginning Blockchain : A Beginner's Guide to Building Blockchain Solutions Bikramaditya Singhal, Gautam

Dhameja, Priyansu Sekhar Panda Apress 2018

2. Mastering Blockchain, Imran Bashir: Packt- Birmingham-Mumbai Third Edition A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, crypto currencies, Ethereum, and more

Reference Books:

1. Blockchain – Blueprint for new Economy Melanie Swan - O'reilly
2. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.
3. Sainul Abideen, Blockchain- ebook, Cybrosys Private Limited.

7KS05 IMAGE PROCESSING (L-3, T-0, C-3)

Course Prerequisite: Calculus, Linear Algebra, Differential Equation

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Image

Processing by being able to do each of the following:

· To introduce and discuss the fundamental concepts and applications of Digital Image Processing.

· To discuss various basic operations in Digital Image Processing.

· To know various transform domains

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to

1. Explain fundamental steps in Image Processing
2. Compare different methods for image transform with its properties
3. Illustrate Image Enhancement in spatial domain
4. Examine Image Enhancement in Frequency Domain
5. Apply various methods for segmenting image and identifying image components
6. Investigate morphological operations to improve the quality of image.

Unit I: Introduction to Image processing: Hours: 7

Fundamental steps in image processing, Components of image processing system, Pixels, coordinate conventions, Imaging Geometry, Spatial Domain, Frequency Domain, sampling and quantization, Basic relationship between pixels, Applications of Image Processing.

Unit II : Image transforms and its properties: Hours: 7

Unitary transform, Discrete Fourier Transform, Discrete Cosine Transform, Walsh Transform, Hadamard Transform.

Unit III: Image Enhancement in spatial domain: Hours: 7

Basic Gray Level Transformation functions – Image Negatives, Log Transformations, Power- Law Transformations. Piecewise-Linear Transformation Functions: Contrast Stretching, Gray Level Slicing, Bit Plane Slicing, Histogram Processing–Equalization, Specification. Basics of Spatial Filtering – Smoothing: Smoothing Linear Filters, Ordered Statistic Filters, Sharpening: Laplacian, Unsharp Masking and High Boost Filtering.

Unit IV: Image Enhancement in Frequency Domain: Hours: 7

Basics of Filtering in Frequency Domain, Filters -Smoothing Frequency Domain Filters : Ideal Low Pass Filter, Gaussian Low Pass Filter, Butterworth Low Pass Filter, Sharpening Frequency Domain Filters: Ideal High Pass Filter, Gaussian High Pass Filter, Butterworth High Pass Filter, Homomorphic Filtering.

Unit V: Image Segmentation: Hours: 7

Pixel-Based Approach- Multi-Level Thresholding, Local Thresholding, Threshold Detection Method, Region-Based Approach- Region Growing Based Segmentation, Region Splitting, Region Merging, Split and Merge, Edge Detection - Edge Operators, Line Detection, Corner Detection.

Unit VI: Morphological Operations: Hours: 7

Basics of Set Theory, Dilation and Erosion - Dilation, Erosion, Structuring Element, Opening and Closing, Hit or Miss Transformation. Representation and Description Representation - Boundary, Chain codes, Polygonal approximation approaches, Boundary segments.

Text Books:

1. A K. Jain, Fundamentals of digital image processing, Prentice Hall of India, 1989.
2. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing (English) 3rd Edition, Pearson India, 2013.

Reference Books:

1. Al Bovik, The Essential Guide to Image Processing, Academic Press, 2009.
2. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing, Analysis, and Machine Vision, Thomson Learning, 2008.
3. S Jayaraman, S Esakkirajan and T Veerakumar, Digital Image Procesing, McGraw Hill Education , 2009.

7KS05 OPTIMIZATION TECHNIQUES (L-3, T-0, C-3)

Course Prerequisite: Mathematics III

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of Optimization Technique by being able to do each of the following:

· To familiarize with optimization techniques using both linear and non-linear programming.
· To study convex optimization though some techniques
· To gain understanding of linear algebra and probability theory

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:

1. Describe statement of an optimization problem
2. Examine linear programming procedures to solve optimization problems.
3. Compare different nonlinear programming methods of optimization
4. Discuss Geometric Programming with different constraint
5. Identify the appropriate optimization technique for the given problem
6. Synthesize algorithms to solve real time optimization problems.

Unit I: Hours: 7

Introduction to Optimization: Introduction, Historical Development, Engineering Applications of Optimization, Statement of an Optimization Problem, Classification of Optimization Problems, Classification Based on the Existence of Constraints.

Classical Optimization Techniques: Introduction, Single-Variable Optimization, Multivariable Optimization with No Constraints, Multivariable Optimization with Equality Constraints, Multivariable Optimization with Inequality Constraints.

Unit II: Hours: 7

Linear Programming I: Simplex Method Introduction, Applications of Linear Programming, Standard Form of a Linear Programming Problem, Geometry of Linear Programming Problems, Definitions and Theorems, Solution of a System of Linear Simultaneous Equations, Pivotal Reduction of a General System of Equations, Motivation of the Simplex Method, Simplex Algorithm, Two Phases of the Simplex Method, Revised Simplex Method, Duality in Linear Programming, Decomposition Principle Sensitivity or Post optimality Analysis, Transportation Problem.

Unit III: Hours: 7

Nonlinear Programming: One-Dimensional Minimization Methods Unimodal Function, ELIMINATION METHODS: Unrestricted Search, Search with Fixed Step Size, Search with Accelerated Step Size, Exhaustive Search, Dichotomous Search, Interval Halving Method, Fibonacci Method, Golden Section Method, Comparison of Elimination Methods, INTERPOLATION METHODS, Quadratic

Interpolation Method, Cubic Interpolation Method, Direct Root Methods, Newton Method, Quasi-Newton Method , Secant Method.

Unit IV: Hours: 7

Nonlinear Programming: Unconstrained Optimization Techniques Introduction, Classification of Unconstrained Minimization Methods, General Approach, Rate of Convergence, Scaling of Design Variables, DIRECT SEARCH

METHODS Random Search Methods, Random Jumping Method, Random Walk Method, Random Walk Method with Direction Exploitation, Advantages of Random Search Methods, Grid Search Method, Univariate Method ,Pattern Directions, Powell's Method, Simplex Method, INDIRECT SEARCH (DESCENT) METHODS Gradient of a Function, Steepest Descent (Cauchy) Method, Conjugate Gradient (Fletcher–Reeves) Method, Newton's Method, Marquardt Method, Quasi-Newton Methods, Davidon–Fletcher–Powell Method, Broyden–Fletcher– Goldfarb–Shanno Method

Unit V: Hours: 7

Nonlinear Programming: Constrained Optimization Techniques Introduction, Characteristics of a Constrained Problem, DIRECT METHODS Random Search Methods, Complex Method, Sequential Linear Programming , Basic Approach in the Methods of Feasible Directions, Zoutendijk's Method of Feasible Directions, Rosen's Gradient Projection Method, Generalized Reduced Gradient Method, Sequential Quadratic Programming, INDIRECT METHODS Transformation Techniques, Basic Approach of the Penalty Function Method, Interior Penalty Function

Method, Convex Programming Problem, Exterior Penalty Function Method, Extrapolation Techniques in the Interior Penalty Function Method, Extended Interior Penalty Function Methods

Unit VI: Hours: 7

Dynamic Programming Introduction, Multistage Decision Processes, Concept of Sub optimization and Principle of Optimality, Computational Procedure in Dynamic Programming, Conversion of a Final Value Problem into an Initial Value Problem, Linear Programming as a Case of Dynamic Programming, Continuous Dynamic Programming Stochastic Programming Introduction, Basic Concepts of Probability Theory, Stochastic Linear Programming, Stochastic Nonlinear Programming, Stochastic Geometric Programming.

Text Book: Engineering Optimization: Theory and Practice, Fourth Edition Singiresu S. Rao Copyright © 2009 by John Wiley & Sons, Inc.

Reference Books:

1. Mokhtar S. Bazaaraa, Hanif D. Shirali and M.C.Shetty, "Nonlinear Programming, Theory and Algorithms", John Wiley & Sons, New York (2004).
2. Kwang Y. Lee, Mohamed A. El-Sharkawi, "Modern heuristic optimization techniques: theory and applications", Kluwer (2008).
3. Hamdy A. Taha, "Operations Research: An Introduction", 8th Edition, Pearson Education (2008).
4. G. V. Reklaitis, A. Ravindran, K. M. Ragsdell, "Engineering Optimization: Methods and Applications", Wiley (2006).
5. Michael C. Bartholomew-Biggs, "Nonlinear optimization with engineering applications", Springer (2008).

7KS06 COMPUTER GRAPHICS – LAB. (P-2, C-1)

Course Prerequisite: Knowledge of C or C++ Programming

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Computer Graphics Lab by being able to do each of the following:

· To acquaint the learner with the basic concepts of Computer Graphics.

· To learn the various algorithms for generating and rendering graphical figures.

· To get familiar with mathematics behind the graphical transformations.

· To understand and apply various methods and techniques regarding projections, animation, shading, illumination and lighting

· To prepare the student for advance areas like Image Processing or Computer Vision or Virtual Reality and professional avenues in the field of Computer Graphics.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:
1. Describe the basic concepts of Computer Graphics.
2. Demonstrate various algorithms for basic graphics primitives.
3. Apply 2-D geometric transformations on graphical objects.
4. Use various Clipping algorithms on graphical objects
5. Explore 3-D geometric transformations, curve representation techniques and projections methods
6. Explain visible surface detection techniques and Animation.

List of Experiments: This is the sample list of Experiments; **minimum 12 experiments** are to be performed covering the entire syllabus. At least two experiments should be beyond syllabi based on learning of syllabi (Apply)
1. Write a program to draw line using DDA algorithm.
2. Write a program to draw line using Bresenham‘s algorithm
3. Write a program to draw circle using Bresenham‘s algorithm
4. Write a program for 2-D transformations, a) Scaling b) Translation c) Rotation
5. Write a program for 3-D transformations, a) Scaling b) Translation c) Rotation
6. Write program to fill polygon using scan line algorithm
7. Write a program to draw the polygons by using the mouse. Choose colors by clicking on the designed color pane. Use window port to draw. Use DDA algorithm for line drawing.
8. Write a program to clip line using following algorithm : Cohen-Sutherland algorithm
9. Write a program to draw following type of curve-Hilbert’s Curve
10. Write a program to draw following type of curve-Koch curve, Bezier curves
11. Write a program to draw inscribed and Circumscribed circles in the triangle as shown as an example below. (Use any Circle drawing and Line drawing algorithms)
12. Write a program to move circle to forward direction.
13. Write a program to draw a cube using in build library function and perform 3D transformations
14. Write a program to fill color in rectangle
15. Write a program to generate Bouncing ball animation using Direct3D/Maya/Blender
16. Write a program to generate snowflake using concept of fractals.
17. Write a program to implement translation, sheer, rotation and scaling transformations on equilateral triangle and rhombus
18. Write program to draw any object such as flower, waves using any curve generation technique
19. Write a program of man walking in rain
20. Write a program to draw a house
21. Write a program for moving a cycle
22. Write a graphics program analog clock
23. Write a program to draw 3-D cube and perform following transformations on it using OpenGL.

7KS07 EMERGING TECHNOLOGY LAB III (P-2, C-1)

7KS07 Emerging Technology Lab III is based on 7KS04 Professional Elective-III. Tentative FOSS Tools & Technology for Practical’s are as follows:
AI : ROS, YARP, MRPT, Gazebo, OROCOS.
DS :RapidMiner, Weka, Scrapy, Pandas
IoT :ThingsBoard, Kinoma, SiteWhere
Cyber Security: Security Onion, LastPass,KeePAss.

7KS08 EMERGING TECHNOLOGY LAB IV(P-2, C-1)

7KS08 Emerging Technology Lab IV is based on 7KS05 Professional Elective-IV. Tentative FOSS Tools & Technology for Practical’s are as follows:
Blockchain: Ethereum,Bigchain DB, Corda

Image Processing:Open CV, SimpleCV, Keras, Caffe
Optimization :Open Eaagles, Repast, Open Simulator.

7KS09 PROJECT AND SEMINAR (P-8, C-4)

Seminar shall bebased on the advanced topic in thefield. It may be related to domain of the project. The seminar should be conducted in seventh semester andevaluated. Each candidate shall submit a seminar report, deliver the seminarand face the viva-voce. The distribution of internal 50 marks shall be asfollows.

1. Seminar report preparation and submission :- 10 marks
2. Seminar delivery/ presentation:- 20 marks
3. Seminar viva-voce:- 10 marks
4. Attendance in all seminar sessions:- 10 marks.

B.E. Sem. VIII (Computer Science & Engineering)

8KS01 OBJECT ORIENTED ANALYSIS AND DESIGN (L-3, T-0, C-3)

Course Prerequisite: Data Structures and algorithms, Basic Mathematics, Geometry, linear algebra, vectors and matrices

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Object- Oriented Analysis and Design by being able to do each of the following:

· To learn the basic concepts of Object-Oriented Analysis and Design, UML, Software Development Processes and Design pattern.

· To study requirement analysis in the Inception phase of software development and relate

· To present Object Oriented Analysis and Design through case studies.

· To introduce design patterns that can be used for development of object-oriented software systems.

· To study UML notation and frequently used UML diagrams for designing Object Oriented software.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:

1. Describe Object Oriented principles, for performing object-oriented analysis and design.
2. Explain the basic concepts of UML, Software Development Processes and Design pattern.
3. Illustrate requirements for developing a software.
4. Create initial domain model & system sequence diagram for use case scenario.
5. Design static and dynamic objects for modeling.
6. Construct UML and Design Patterns for developing object-oriented software.

Unit I: Hours: 07

Introduction to Object Oriented Analysis and Design: Analysis and Design, Object-Oriented Analysis and Design; UML, Iterative, Evolutionary and Agile: UP, Iterative and Evolutionary Development, Waterfall Lifecycle, Iterative and Evolutionary Analysis and Design, Risk-Driven and Client-Driven Iterative Planning, Agile Methods and Attitudes, Agile Modeling, Agile UP, UP Phases, UP Disciplines.

Unit II: Hours:07

Defining Inception: Inception, Artifacts Start in Inception, Evolutionary requirements: Requirements, Evolutionary vs. Waterfall Requirements, Types and Categories of Requirements, Requirements Organized in UP Artifacts Use cases: Actors, Scenarios and Use Case, Use Cases and the Use-Case Model, Importance of Use Cases, Three Kinds of Actors, Three Common Use Case Format, Sections Mean, Take an Actor and Actor- goal perspective, Use Case Diagrams, Activity Diagrams

Unit III: Hours:07

Domain Models: Domain Model, Need of Create a Domain Model, create a Domain Model, Conceptual Classes, Sketching a Class Diagram, Common Mistake with Attributes vs. Classes, Associations, Attributes. System Sequence Diagrams: System Sequence Diagrams, Need of SSD, Relationship between SSDs and Use Cases, Naming System Events and Operations, Model SSDs Involving Other External Systems, Process: Iterative and Evolutionary SSDs, Operation Contracts.

Unit IV: Hours:07

Logical Architecture and UML Package Diagrams: Logical Architecture, Layers, Software Architecture, UML Package Diagrams, Design with Layers, Benefits of Using Layers

On to Object Design: Designing Objects: Static and Dynamic Modeling, The Importance of Object Design Skill over UML Notation Skill UML Interaction Diagrams: Sequence and Communication Diagrams, Common UML Interaction Diagram Notation, Basic Sequence Diagram Notation, Basic Communication Diagram Notation. UML Class Diagram: Common Class Diagram Notation, Design Class Diagram, Attribute Text and Association Lines, Notes, Comments, Constrains and Method Bodies, Operations and Methods, Keywords, Stereotypes, Profiles and Tags

Unit V: Hours:07

GRASP: Designing Objects with Responsibilities: Object Design: Example Inputs, Activities and Outputs, Responsibilities and Responsibility-Driven Design, GRASP: A Methodological Approach to Basic OO Design, the Connection between Responsibilities, GRASP and UML Diagrams, Patterns, A Short Example of Object Design with GRASP Designing for Visibility: Visibility between Objects Mapping Designs to Code: Creating Class Definitions from DCDs, Creating Methods from Interaction Diagrams, Collection Classes in Code

Unit VI: Hours:07

Applying GoF Design Patterns: Adapter(GoF), Factory, Singleton(GoF), Strategy (GoF), Composite (GoF) and Other Design Principles, Façade (GoF), Observer (GoF). UML State Machine Diagrams and Modeling: Event, State and Transition, Apply State Machine Diagrams, More UML State Machine Diagram Notation, State Machine Diagrams in UP. Relating Use Cases: The include Relationship, The extend Relationship, The Generalize Relationship, Use Case Diagrams.

Text Books:

1. Craig Larman: "Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development, Third Edition, Addison Wesley Professional.
2. Erich Gamma et al., Design Patterns, Elements of Reusable OO Software, Addison-Wesley.

Reference Books:

1. Blaha, Rumbaugh: "Object Oriented Modeling and Design with UML" (2/e) Pearson Education.
2. Arlow, Jim, "UML and the Unified Process", Pearson Education.
3. Dathan, Ramnath: "Object Oriented Analysis, Design & Implementation, "OUP.
4. McRobb& Farmer: "Object Oriented System Analysis & Design" Mc Graw Hill.
5. Booch, Rumbaugh & Jacobson: " The UML User guide" Pearson Education.
6. Whitten & Bentley: "System Analysis & Design Methods" Tata McGraw Hill.
7. Booch: "Object Oriented Analysis & Design with Applications", Pearson Education.

8KS02 PROFESSIONAL ETHICS AND MANAGEMENT (L-3, T-0, C-3)

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Professional Ethics and Management by being able to do each of the following:

· To enable the students to create an awareness of engineering and professional ethics

· To instill moral, social values and appreciate the rights of others

· To regulate the student's behavior in a professional environment

· To conscious about the impact of non-ethical engineering decisions
· To comprehend ‘mind and desire control’ needs for being ethical

Course Outcomes (Expected Outcomes): On completion of the course, the students will be able to:

1. Relate ethical and non-ethical situations
2. Outline ethics in the society & environment
3. Examine the moral judgment & correlate the concepts in addressing the ethical dilemmas
4. Identify risk and safety measures in various engineering fields
5. Justify ethical issues related to engineering responsibilities and rights
6. Synthesize cognitive skills in solving social problems

Unit I: Introduction to Ethics Hours: 07

Senses of Engineering and professional ethics, Engineering profession & its view, Ethical issues for engineers, distinction between ethics, morals and laws, opinions vs. judgments, Ethical theories: utilitarianism, duty, right, virtue; Cost-benefit analysis in engineering, McCuen’s ethical dimensions, IEEE: Code of conducts & Code of ethics

Unit II: Professional Practices in Engineering Hours: 07

Professional attributes, Difference in engineering and other professions; Ethical dilemma: right-wrong or betterworse; Code of ethics for engineers in India: need and its roles; abuse of codes, ethical relativism, well-being and profession, Ethics as Design - Doing Justice to Moral Problems, Kohlberg’s theory – Gilligan’s theory.

Unit III: Central Professional Responsibilities of Engineers Hours: 07

Confidentiality and Proprietary Information, Conflict of interest, Competitive bidding, rights of Engineers: fundamental, professional conscience, conscientious refusal, professional recognition, employee, privacy; types of conflict of interest, avoiding conflict of interest, competitive bidding, situations for conflict of interest, ethical corporate climate & its features.

Unit IV: Intellectual Property Rights and Ethics Hours: 07

Patent: IP chain of activities, IP as intangible property, protection offered by patent, right of patent owner; Trademarks (TM): purpose, what can be registered under trademark, categories of TM, industrial design, geographical indications; Copyright & related rights: advances in technology and copyright, benefits, World IP organization, TRIPS & WTO.

Unit V: Computers, Software and Digital Information Hours: 07

Emergence of Computer ethics, issues in Computer ethics: distribution of power issues, property issues, issues of privacy, professional issues, Computer crimes, Computer Software and Digital Information: Characteristics of digital information, s/w as IP, and challenges in information age, IEEE code of conduct and code of ethics.

Unit VI: Responsibilities and Management Hours: 07

Responsibility for the Environment, Engineering as Social Experimentation, Safety and Risk management, IT Professional relationship management with: Employers, Clients, Suppliers, IT Users, other professionals, and society at large.

Text Books:

1. Prof. Susmita Mukhopadhyay, ‘Ethics in Engineering Practice’ IIT Kharagpur
2. Mike Martin and Roland Schinzinger, ‘Ethics in Engineering’, Tata McGraw Hill, New York, 2005

Reference Books:

1. Charles E. Harris, Michael S. Pritchard and Michael J. Rabins, ‘Engineering Ethics – Conceptsand Cases’, Cengage Learning, 2009 & Thompson Learning, 2000
2. Govindarajan M., Natarajan,‘Engineering Ethics’, Prentice Hall of India, New Delhi, 2004
3. Stephen Byars, ‘Business Ethics’, USC Marshal School of Business Kurt Stanberry, University of Houston (https://openstax.org/details/books/business-ethics)

8KS03 VIRTUAL AND AUGMENTED REALITY (L-3, T- 0, C-3)

Course Prerequisite: Basics of Computers & Multimedia

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Virtual and Augmented Reality by being able to do each of the following:

· To introduce the relevance of this course to the existing technology through demonstrations, case studies and applications with a futuristic vision along with socio-economic impact and issues

· To understand virtual reality, augmented reality and using them to build Biomedical engineering applications

· To know the intricacies of these platform to develop PDA applications with better optimality

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:

1. Describe Virtual reality & its applications.
2. Discuss virtual reality world and types.
3. Examine geometry of virtual world and the physiology of human vision
4. Investigate Visual Perception, Motion and Tracking
5. Inspect Physics of Sound and the Physiology of Human Hearing.
6. Explain Augmented reality & examples based on Augmented reality

Unit I: Hours: 07

Introduction to Virtual Reality: Defining Virtual Reality, History ofVR, Human Physiology and Perception, Key Elements of Virtual Reality Experience, Virtual Reality System, Interface to the Virtual World-Input & output- Visual, Aural & Haptic Displays, Applications of Virtual Reality.

Unit II: Hours: 07

Representing the Virtual World: Representation of the Virtual World, Visual Representation in VR, Aural Representation in VR and Haptic Representation in VR

Unit III: Hours: 07

The Geometry of Virtual Worlds &The Physiology of Human Vision: Geometric Models, Changing Position and Orientation, Axis- Angle Representations of Rotation, Viewing Transformations, Chaining the Transformations, Human Eye, eye movements & implications for VR.

Unit IV: Hours: 07

Visual Perception, Motion & Tracking: Visual Perception -Perception of Depth, Motion, & Color, Ray Motion in Real and Virtual Worlds- Velocities and Accelerations, Tracking 2D & 3D Orientation, Tracking Position and Orientation.

Unit V: Hours:07

Interaction & Audio: Interaction - Motor Programs and Remapping, Locomotion, Manipulation, Social Interaction. Audio -The Physics of Sound, The Physiology of Human Hearing, Auditory Perception, Auditory Rendering.

Unit VI: Hours: 07

Basics of Augmented Reality: Introduction to Augmented Reality , Examples based on Augmented reality , Mixed Reality Continuum ,Computer Vision for Augmented Reality , Confluence of Virtual Reality and Augmented Reality , Requirements of AR Authoring ,Taking AR Outdoors.

Text Books:

1. M. LaValle, “Virtual Reality, Steven”, Cambridge University Press, 2016.
2. Augmented Reality: Principles and Practice (Usability) by Dieter Schmalstieg & Tobias Hollerer, Pearson Education (US), Addison-Wesley Educational Publishers Inc, New Jersey, United States, 2016. ISBN: 9780321883575

Reference Books:

1. William R Sherman and Alan B Craig, “Understanding Virtual Reality”, Interface, Application and Design, (The Morgan Kaufmann Series in Computer Graphics)”. Morgan Kaufmann Publishers, San Francisco, CA, 2002
2. Alan B Craig, William R Sherman and Jeffrey D Will, “Developing Virtual Reality Applications: Foundations of Effective Design”, Morgan Kaufmann, 2004
3. Burdea, Grigore C and Philippe Coiffet, “Virtual Reality Technology”, Wiley Inter science, India, 2008
4. Alan B. Craig, Understanding Augmented Reality, Concepts and Applications, Morgan Kaufmann, 2013.

5. Gerard Jounghyun Kim, “Designing Virtual Systems: The Structured Approach”, 2005.
6. Doug A Bowman, Ernest Kuijff, Joseph J LaViola, Jr and Ivan Poupyrev, “3D User Interfaces, Theory and Practice”, Addison Wesley, USA, 2005.
7. Oliver Bimber and Ramesh Raskar, “Spatial Augmented Reality: Merging Real and Virtual Worlds”, 2005
8. Jason Jerald - The VR Book: Human-Centred Design for Virtual Reality. Association for Computing Machinery and Morgan and Claypool, New York, NY, USA.
9. Dieter Schmalstieg and Tobias Hollerer - Augmented Reality: Principles and Practice (Usability), Pearson Education (US), Addison-Wesley Educational Publishers Inc, New Jersey, United States, 2016.
1. 8. Steve Aukstakalnis - Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR (Usability), Addison-Wesley Professional; 1st edition, 2016.
10. Robert Scoble and Shel Israel - The Fourth Transformation: How Augmented Reality and Artificial Intelligence Will Change Everything, Patrick Brewster Press; 1st edition, 2016.
11. Tony Parisi - Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile, OReilly Media; 1st edition, 2015.
12. Programming 3D Applications with HTML5 and WebGL: 3D Animation and Visualization for Web Pages, Tony Parisi, OReilly Media; 1st edition, 2014.
13. John Vince - Virtual Reality Systems, Addison Wesley, 1995.
14. Howard Rheingold - Virtual Reality: The Revolutionary Technology and how it Promises to Transform Society, Simon and Schuster, 1991.

Supplementary Resources:

1. http://lavalle.pl/vr/book.html

Mapped with MOOCS/other Courses:

1. https://nptel.ac.in/courses/106/106/106106138/
2. https://nptel.ac.in/courses/106105195/13
3. https://www.coursera.org/learn/introduction-virtual-reality.

8KS03 PE V: (ii) MACHINE LEARNING AND AI

Course Prerequisite: Basic Mathematics, Linear algebra, Vectors and matrices, Data Science & Statistics

Course Objectives:

Throughout the course, students will be expected to demonstrate their understanding of Machine Learning and AI by being able to do each of the following:

- To introduce students to the basic concepts and techniques of Machine Learning.
- To have a thorough understanding of the Supervised and Unsupervised learning techniques
- To study the various probability-based learning techniques
- To understand neural network

Course Outcomes (Expected Outcome):

On completion of the course, the students will be able to:

1. Describe Machine learning and its types.
2. Discuss Bayesian Decision Theory and Parametric Methods
3. Illustrate Multivariate and Dimensionality Reduction methods.
4. Categorize Non-Parametric methods.
5. Justify discrimination techniques in Machine learning.
6. Synthesize Neural network using Multilayer Perceptron.

Unit I: (Hours7)

Multilayer Perceptions: Introduction: Understanding the Brain, Neural Networks as a Paradigm for Parallel Processing; The Perceptron, Training a Perceptron, Learning Boolean Functions, Multilayer Perceptron’s, MLP as a Universal Approximator, Back propagation Algorithm: Nonlinear Regression, Multiple Hidden Layers.

Unit II: (Hours7)

Introduction: What Is Machine Learning Examples of Machine Learning Applications, Learning Associations, Classification, Regression, Unsupervised Learning, Reinforcement Learning Supervised Learning, Vapnik Chervonenk is Dimension, Dimensions of a Supervised Machine Learning Algorithm.

Unit III: (Hours7)

Bayesian Decision Theory: Introduction, Classification, Losses and Risks, Discriminant Functions, Association Rules Parametric Methods: Introduction, Maximum Likelihood Estimation, Bernoulli Density, Evaluating an Estimator: Bias and Variance, Tuning Model Complexity: Bias/Variance Dilemma, Model Selection Procedures

Unit IV: (Hours7)

Multivariate Methods: Multivariate Data, Parameter Estimation, Estimation of Missing Values, Multivariate Normal Distribution, Multivariate Classification, Tuning Complexity, Dimensionality Reduction: Introduction, Subset Selection, Principal Component Analysis, Feature Embedding, Factor Analysis.

Unit V: (Hours7)

Clustering: Introduction, Mixture Densities, k-Means Clustering, Expectation-Maximization Algorithm, Mixtures of Latent Variable Models, Supervised Learning after Clustering, Spectral Clustering, Hierarchical Clustering, Choosing the Number of Clusters, Condensed Nearest Neighbor, Distance-Based Classification, Outlier Detection.

Unit VI: (Hours7)

Decision Trees: Introduction, Univariate Trees, Classification Trees, Regression Trees, Pruning, Rule Extraction from Trees, Learning Rules from Data, Multivariate Trees. Linear Discrimination: Introduction, Generalizing the Linear Model.

Text Book:

Ethem Alpaydin, -Introduction to Machine Learning 3e (Adaptive Computation and MachineLearning Series), Third Edition, MIT Press, 2014

Reference Books:

1. Stephen Marsland, -Machine Learning – An Algorithmic Perspective‖, Second Edition, Chapman and Hall/CRC

Machine Learning and Pattern Recognition Series, 2014.

2. Tom M Mitchell, -Machine Learning‖, First Edition, McGraw Hill Education, 2013.

3. Peter Flach,Machine Learning: The Art and Science of Algorithms that Make Sense of Data‖, First Edition, Cambridge University Press, 2012.

8KS03 WIRELESS SENSOR NETWORKS (L-3, T-0, C-3)

Course Prerequisite: Computer Networks, Internet of Things, Sensors and Actuators

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Wireless Sensor Network by being able to do each of the following:

· To understand the fundamentals of wireless sensor networks and its application to critical real time scenarios.

· To study the various protocols at various layers and its differences with traditional protocols.

· To understand the issues pertaining to sensor networks and the challenges involved in managing a sensor network.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:

1. Describe Network of Wireless Sensor Nodes
2. Explain Node Architecture and Physical Layer.
3. Discuss Medium Access Control and its related properties.
4. Analyze the protocols and algorithms used at different network protocollayers in sensor systems.
5. Compare different power management techniques and clocks and the Synchronization problems.
6. Explain time synchronization and its problems.

Unit I: Hours:07

Network of Wireless Sensor Nodes- Definitions and Background, Sensing and Sensors, Wireless Sensor Networks, Challenges and Constraints, Energy, Self-Management, Wireless Networking, Decentralized Management, Design Constraints, Security, Other Challenges. Applications: Structural Health Monitoring, Traffic Control, Health Care, Pipeline Monitoring, Precision Agriculture, Active Volcano, Underground Mining.

Unit II: Hours:07

Node Architecture: The Sensing Subsystem, The Processor Subsystem, Communication Interfaces, Prototypes. Physical Layer: Basic Components, Source Encoding, Channel Encoding, Modulation, Signal Propagation.

Unit III: Hours:07

Medium Access Control: Contention-Free Medium Access, Contention-Based Medium Access, Wireless MAC Protocols, Characteristics of MAC Protocols in Sensor Networks, Contention-Free MAC Protocols, Contention- Based MAC Protocols, Hybrid MAC Protocols.

Unit IV: Hours:07

Network Layer: Routing Metrics, Flooding and Gossiping, Data-Centric Routing, Proactive Routing, On-Demand Routing, Hierarchical Routing, Location-Based Routing, QoS-Based Routing Protocols.

Unit V: Hours:07

Power Management: Local Power Management Aspects, Dynamic Power Management, Conceptual Architecture. Time Synchronization: Clocks and the Synchronization Problem, Time Synchronization in Wireless Sensor Networks, Basics of Time Synchronization, Time Synchronization Protocols.

Unit VI: Hours:07

Localization: Ranging Techniques, Range-Based Localization, Range-Free Localization, Event-Driven Localization. Security: Fundamentals of Network Security, Challenges of Security in Wireless Sensor Networks, Security Attacks in Sensor Networks, Protocols and Mechanisms for Security, IEEE 802.15.4 and Zig Bee Security.

Text Book:

Fundamentals of Wireless Sensor Networks: Theory and Practice / Waltenegus Dargie, Christian Poellabauer, 2010 John Wiley & Sons Ltd.

Reference Books:

1. .Wireless sensor networks: technology, protocols, and applications by Kazem Sohraby, Daniel Minoli, TaiebZnati, Copyright _ 2007 by John Wiley & Sons, Inc.
2. Wireless Sensor Network Designs by Anna Hac, John Wiley & Sons Ltd.
3. Wireless Sensor Networks by Ian F. Akyildiz, Mehmet Can Vuran, 2010 John Wiley & Sons Ltd.
4. Wireless Sensor Networks: An Information Processing Approach by Feng Zhao, Leonidas J. Guibas, The Morgan Kaufmann Series in Networking.

8KS03 Prof. Elect. V (iv) SYSTEM & SOFTWARE SECURITY

Course Prerequisite: Networking, Operating System, Basics of Cyber Security & Cryptography

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of System and Software Security by being able to do each of the following:

1. To provide an in-depth study of concepts and threats in computer security.
2. To provide knowledge of common vulnerabilities, attack mechanisms and methods against computer and information system
3. To familiarize security issues at various levels such as operating systems and databases.
4. To provide the study of vulnerability issues and its counter measures at advanced application such as networks and Clouds.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:

1. Relate malicious and non-malicious attacks.

2. Outline web common vulnerabilities, attack mechanisms and methods against computer and information systems.
3. Apply relevant methods for security modeling and analysis of Operating System.
4. Investigate a secure network by monitoring and analyzing the nature of attacks.
5. Explain cryptography, intrusion detection and firewall system
6. Implement different security solutions at various levels such as operating systems, databases and clouds.

Unit I: (Hours7)

Programs and Programming: Unintentional (Non malicious) Programming: Buffer Overflow, Incomplete Mediation, Time-of-Check to Time-of Use, Undocumented Access Point, Off-by-One Error, Integer Overflow, Un-terminated Null-Terminated String, Parameter Length, Type and Number, Unsafe Utility Program, Race Condition. Malicious Code: Malware: Viruses, Trojan Horses, and Worms, Technical Details: Malicious Code. Countermeasures: Countermeasures for Users, Countermeasures for Developers, Countermeasure Specifically for Security, Countermeasures that Don’t Work.

Unit II: (Hours7)

The Web: Browser Attacks: Browser Attack Types, How Browser Attacks Succeed: Failed Identification and Authentication. Web Attacks Targeting Users: False or Misleading Content, Malicious Web Content, Protecting Against Malicious Web Pages. Obtaining User or Website Data: Code Within Data, Website Data: A User’s Problem, Too Foiling Data Attacks. Email Attacks: Fake Email, Fake Email Messages as Spam, Fake (Inaccurate) Email Header Data, Phishing, Protecting Against Email Attacks.

Unit III: (Hours7)

Operating System: Security in Operating Systems: Operating System Structure, Security Features of Ordinary Operating Systems, Protected Objects, Operating System Tools to Implement Security Functions. Security in the Design of Operating Systems: Simplicity of Design, Layered Design, Kernelized Design, Reference Monitor, Correctness and Completeness, Secure Design Principles, Trusted Systems, Trusted System Functions, The Results of Trusted Systems Research Rootkit: Phone Rootkit, Rootkit Evades Detection, Rootkit Operates Unchecked, Sony XCP Rootkit, TDSS Rootkits, Other Rootkits.

Unit IV: (Hours7)

Networks: Network Concepts, Threats to Network Communications: Interception: Eavesdropping and Wiretapping, Modification, Fabrication: Data Corruption Interruption: Loss of Service, Port Scanning, Vulnerability. Wireless Network Security: Vulnerabilities in Wireless Networks, Failed Countermeasure: WEP (Wired Equivalent Privacy), Stronger Protocol Suite: WPA (Wi-Fi Protected Access) Denial of Service, Cryptography in Network Security Browser Encryption, Onion Routing, IP Security Protocol Suite (IPsec), Virtual Private Networks, System Architecture. Firewalls: Firewall, Design of Firewalls, Types of Firewalls, Personal Firewalls Comparison of Firewall Types, Example Firewall Configurations.

Unit V: (Hours7)

Database: Security Requirements of Databases: Integrity of the Database, Element Integrity, Auditability, Access Control, User Authentication, Availability, Integrity / Confidentiality/Availability. Reliability and Integrity: Two- Phase Update Redundancy/Internal Consistency, Recovery, Concurrency/Consistency. Database Disclosure: Sensitive Data, Types of Disclosures, Preventing Disclosure: Data Suppression and Modification, Security Versus Precision Data Mining and Big Data: Data Mining, Big Data.

Unit VI: (Hours7)

Cloud Computing: Cloud Computing Concepts: Service Models, Deployment Models. Risk Analysis: Cloud Provider Assessment, Switching Cloud Providers, Cloud as a Security Control. Cloud Security Tools and Techniques: Data Protection in the Cloud, Cloud Application Security, Logging and Incident Response. Cloud Identity Management: Security Assertion Markup Language OAuth: OAuth for Authentication. Securing IaaS.

Text Book:

Security in Computing, Charles P. Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies, Fifth Edition, Prentice Hall, 2015.

Reference Books:

1. Computer Security: Principles and Practice, William Stallings and Lawrie Brown, Third Edition, Pearson Prentice Hall
2. Web Technologies: TCP/IP, Web/Java Programming, and Cloud Computing Achyut S. Godbole,Tata McGraw- Hill Education, 2013
3. Cryptography and Network Security Principles and Practices, William Stallings, Seventh Edition, Pearson
4. Michael T. Goodrich and Roberto Tamassia, Introduction to Computer Security, Addison Wesley, 2011

8KS04 DISTRIBUTED LEDGER TECHNOLOGY (L-3, T-0, C-3)

Course Prerequisite: Data structures and Algorithms, Design and Analysis of Algorithms, Discrete Mathematics and basic knowledge of Cryptography

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Distributed Ledger Technology by being able to do each of the following:

· To develop an understanding of the requirements for electronic payment systems

· To understand key cryptographic constructs, economic incentive mechanisms and distributed algorithms underpinning crypto currencies such as Bitcoin and Ethereum

· To develop a basic facility with programming smart contracts on one crypto currency platform.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:

1. Describe basic knowledge of Distributed Ledger Technologies
2. Outline Analytical Framework for Distributed ledger technology
3. Use Cryptographic method for ledgers.
4. Explain knowledge of Bitcoin
5. Inspect Bitcoin cryptocurrency mechanisms
6. Synthesize bitcoin mining process.

Unit I: Hours:07

Distributed ledger technology: Introduction, Background, Technical design elements, Institutional design elements: Operation of the arrangement, Access to the arrangement (unrestricted or restricted)

Unit II: Hours:07

Analytical framework: Understanding the arrangement, Potential implications for efficiency, Potential implications for safety, Potential broader financial market implications

Unit III: Hours:07

Introduction to Cryptography & Cryptocurrencies: Cryptographic Hash Functions, SHA-256, Hash Pointers and Data Structures, Digital Signatures, Public Keys as Identities, A Simple Cryptocurrency

Unit IV: Hours:07

Bitcoin: Centralization vs. Decentralization, Distributed consensus, Consensus without identity using a block chain, Incentives and proof of work

Unit V: Hours:07

Mechanics of Bitcoin: Bitcoin transactions, Bitcoin Scripts, Applications of Bitcoin scripts, Bitcoin blocks, Bitcoin network. How to Store and Use Bitcoins, Simple Local Storage, Hot and Cold Storage, Splitting and Sharing Keys, Online Wallets and Exchanges, Payment Services, Transaction Fees, Currency Exchange Markets

Unit VI: Hours:07

Bitcoin Mining: The task of Bitcoin miners, Mining Hardware, Energy consumption and ecology, Mining pools, Mining incentives and strategies.

Text Book:

Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction: Andrew Miller, Arvind Narayanan, Edward Felten, Joseph Bonneau, and Steven Goldfeder. Princeton University.

Reference Books:

1. Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained, 2nd Edition
2. Distributed ledger technology in payment, clearing and settlement - An analytical framework
3. Dr. Gavin Wood, "ETHEREUM: A Secure Decentralized Transaction Ledger,"Yellow paper.2014.
4. Nicola Atzei, Massimo Bartoletti, and TizianaCimoli, A survey of attacks on Ethereum smart contracts.

8KS04 Prof. Elect. VI (ii) MULTIMEDIA COMPUTING

Course Prerequisite: Computer Network, Image Processing

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Multimedia Computing by being able to do each of the following:

1. To learn and understand technical aspect of Multimedia Computing.
2. To understand the standards available for different audio video and text applications.
3. To Design and develop various Multimedia Systems applicable in real time.
4. To learn various multimedia compression algorithms.
5. To understand various networking aspects used for multimedia applications.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to

1. Describe technical aspect of Multimedia Computing.
2. Compare various file formats for audio, video and text media.
3. Examine lossless data compression techniques in real time.
4. Illustrate lossy data compression techniques in real time scenario
5. Investigate video compression technique
6. Construct various networking protocols for multimedia applications.

Unit I: Introduction: (Hours7)

Fundamental concepts in Text and Image: Multimedia and hypermedia, World Wide Web, overview of multimedia software tools. Graphics and image data representation graphics/image data types, file formats, Color in image and video: color science, color models in images, color models in video.

Unit II: Video and Digital Audio: (Hours7)

Fundamental concepts in video and digital audio: Types of video signals, analog video, digital video, digitization of sound, MIDI, quantization and transmission of audio.

Unit III: Data Compression-I: (Hours7)

Multimedia data compression I: Lossless compression algorithm: Run-Length Coding, Variable Length Coding, Dictionary Based Coding, Arithmetic Coding, Lossless Image Compression.

Unit IV: Data Compression-II: (Hours7)

Multimedia data compression II: Lossy compression algorithm: Quantization, Transform Coding, Wavelet-Based Coding, Embedded Zero tree of Wavelet Coefficients Set Partitioning in Hierarchical Trees (SPIHT).

Unit V: Video Compression: (Hours7)

Basic Video Compression Techniques: Introduction to video compression, video compression based on motion compensation, search for motion vectors, MPEG, Basic Audio Compression Techniques.

Unit VI: Multimedia Networks: (Hours7)

Basics of Multimedia Networks, Multimedia Network Communications and Applications: Quality of Multimedia Data Transmission, Multimedia over IP, Multimedia over ATM Networks, Transport of MPEG-4, Media-on-Demand (MOD).

Text Book: Fundamentals of Multimedia by Ze-Nian Li and Mark S. Drew Pearson Education.

Reference Books:

1. Digital Multimedia, Nigel chapman and jenny chapman, Wiley-Dreamtech
2. Macromedia Flash MX Professional 2004 Unleashed, Pearson.
3. Multimedia and communications Technology, Steve Heath, Elsevier (Focal Press).
4. Multimedia Applications, Steinmetz, Nahrstedt, Springer.
5. Multimedia Technology and Applications, David Hilman, Galgotia.

8KS04 MODELLING & SIMULATION (L-3, T-0, C-3)

Course Prerequisite: Familiarity with Linear Algebra, Probability and Statistics, Discrete structures, graph theory, Object-oriented design and programming.

Course Objectives: Throughout the course, students will be expected to demonstrate their understanding of Modelling & Simulation by being able to do each of the following:

· To understand the basic concepts in modeling and simulation
· To introduce the simulation and modeling techniques
· To introduce basic simulation and modeling skills with respect to carrying out research projects using any simulation method on the computer.

Course Outcomes (Expected Outcome): On completion of the course, the students will be able to:

1. Describe System models & system modelling.
2. Explain continuous system methods of obtaining solutions.
3. Illustrate the need of simulation and mathematical modeling
4. Examine simulation of Queuing System and PERT network.
5. Inspect experimentation of Simulation.
6. List different special purpose languages use for continuous and discrete systems

Unit I: Hours:7

System Models and System studies: Basic concepts of systems and system modeling static and dynamic/physical and mathematical models-principles used in modeling-corporate models-analysis, design and postulation of system.

Unit II: Hours:7

Basic Concepts and continuous system: Techniques used - distributed lag models and cobweb models continuous system Model-Analytical equations &methods of obtaining solutions–analog and hybrid computers and simulations CSSLS examples of different continuous system.

Unit III: Hours:7

System dynamics, probability concepts and basic principles of discrete simulation Growth and decay models system dynamics diagrams examples - stochastic process-probability functions and their evaluation –random number generation-rejection method-comparison of Monte-Carlo method and stochastic simulation – examples

Unit IV: Hours:7

Simulation of Queuing system and PERT Network, Simulation of Queuing system: Rudiments of queuing theory, simulation of a single serve queue, simulation of a two-server queue, simulation of more general queues, Simulation of a PERT Network: Network model of a project, Analysis of an activity network, critical path.

Unit V: Hours:7

Simulation of Inventory Control & Forecasting Design and Evaluation of Simulation Experiments Inventory Control and Forecasting, Elements of inventory theory, more Complex inventory models simulation example= 1 Generation of Poisson and Erlanger variates, Simulation example-2 Forecasting and regression Analysis. Design and Evaluation of simulation Experiments: Length of Simulation runs, Variance reduction techniques, Experimental layout, Validation summary and conclusion.

Unit VI: Hours:7

Simulation of Languages and Introduction to GPSS, Different special purpose languages use for continuous and discrete systems and comparison, factors affecting the selection of discrete system simulation languages-comparison of GPSS sans SIMSCRIPT.A detailed study of GPSS with examples.

Text Books:

1. Geoffrey Gordon, System Simulation, PHI Learning/Pearson.
2. Narsingh Deo, System Simulation with Digital Computer, PHI Learning/Pearson.

Reference Books:

1. Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol, P. Shahabudeen, Discrete-Event System Simulation, Fourth Edition, Pearson Publication.
2. Sheldon Ross, Simulation , Academic Press.
3. Law & Kelton, Simulation Modeling & Analysis, Tata McGraw Hill
4. Kai Velten, Mathematical Modeling and Simulation: Introduction for Scientists and Engineers, Wiley.
5. Shannon, R.E. Systems simulation, The art and science, Prentice Hall, 1975.
6. Thomas J. Schriber, Simulation using GPSS, John Wiley, 1991.

8KS05 EMERGING TECHNOLOGY LAB V (P-2, C-1)

8KS05 Emerging Technology Lab V is based on 8KS03 Professional Elective-V. Tentative FOSS Tools & Technology for Practical's are as follows:
AI :Google's ARCore, AR.js, ARToolKit, DroidAR, Brio, Adobe Aero
DS :R Studio, Orange, D3.js, Ggplot2, Jupyter Notebooks
IoT :DSA,Thinger,RIOT, OpenRemote,Anjay
Cyber Security: Wireshark, Burp Suit, Nessus.

8KS06 EMERGING TECHNOLOGY LAB VI (P-2, C-1)

8KS06 Emerging Technology Lab V is based on 8KS04 Professional Elective-VI. Tentative FOSS Tools & Technology for Practical's are as follows:
Blockchain: Hyperledger, HydraChain, MultiChain, Elements
Image Processing:Google Colab, GPUImage, Cuda, Aforge/Accord.NET
Optimization: OR-Tools, Locust.io, httperf, Apache JMeter, Siege.

8KS07 PROJECT & SEMINAR (P-12, C-6)

The student batch size for project may be preferably 04. The project shall be internally evaluated (for 75 Internal Marks) inthree phases based on theprogress of the project work. Each phase shallbe internally evaluated for 25 marks as follows:
Phase I: - Problem Definition and Design
Phase II: - Problem Implementation and Testing
Phase III: - Project Demonstration & Report submission.
The external evaluation of the project shall be based ondemonstration of the project and viva-voce.