Bansilal Ramnath Agarwal Charitable Trust’s

Vishwakarma Institute of Technology
(An Autonomous Institute affiliated to University of Pune)

Structure & Syllabus of

B.Tech. (Information Technology)

Pattern ‘D19’
Effective from Academic Year 2019-20

Prepared by: - Board of Studies in Information Technology

Approved by: - Academic Board, Vishwakarma Institute of Technology, Pune

Signed by

Chairman – BOS       Chairman – Academic Board
## Content

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>A</strong> Program Educational Objectives and Program Outcome of B.Tech</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>(Information Technology)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Course Structure - Module VII/VIII</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>Course Syllabi for Courses - Module VII/VIII</td>
<td>7</td>
</tr>
<tr>
<td>3.1</td>
<td>IT4085 Distributed Computing</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>IT4018 Convergence Technology</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>IT4006 Artificial intelligence</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>IT4190 Financial Technology</td>
<td>16</td>
</tr>
<tr>
<td>3.3</td>
<td>IT4012 Image Processing</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>IT4019 Software Design Methodology</td>
<td>20</td>
</tr>
<tr>
<td>3.4</td>
<td>IT4115/IT4116 Major Project 1/ Major Project 2</td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>IT4072/IT4117/IT4032/IT4030/IT4031 Summer Internship/Industry Internship/International Internship/Research Assistant/Project Internship</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>B.Tech Elective</strong></td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>DAA</td>
<td>24</td>
</tr>
<tr>
<td>4.2</td>
<td>Data Science</td>
<td>27</td>
</tr>
<tr>
<td>4.3</td>
<td>Artificial Neural Network</td>
<td>30</td>
</tr>
</tbody>
</table>
Program Educational Objectives (PEO)
B.Tech (Information Technology)
List of Programme Education Objectives [PEO] and Programme Outcomes [PO]

<table>
<thead>
<tr>
<th>PEO</th>
<th>PEO Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEO1</td>
<td>Preparation: Demonstrate application of sound engineering foundations to be a committed technology workforce</td>
</tr>
<tr>
<td>PEO2</td>
<td>Core competence: Apply mathematical and computing theory knowledge base to provide realistic computer engineering solutions</td>
</tr>
<tr>
<td>PEO3</td>
<td>Breadth: Exhibit problem solving skills and engineering practices to address problems faced by industry with innovative methods, tools and techniques</td>
</tr>
<tr>
<td>PEO4</td>
<td>Professionalism: Adopt professional and ethical practices adopting effective guidelines to acquire desired soft skills in societal and global context</td>
</tr>
<tr>
<td>PEO5</td>
<td>Learning Environment: Aim for continuing education and entrepreneurship in emerging areas of computing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PO</th>
<th>PO Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO1</td>
<td>Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.</td>
</tr>
<tr>
<td>PO2</td>
<td>Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.</td>
</tr>
<tr>
<td>PO3</td>
<td>Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.</td>
</tr>
<tr>
<td>PO4</td>
<td>Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.</td>
</tr>
<tr>
<td>PO5</td>
<td>Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.</td>
</tr>
<tr>
<td>PO6</td>
<td>The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.</td>
</tr>
<tr>
<td>PO7</td>
<td>Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for</td>
</tr>
<tr>
<td>PO8</td>
<td><strong>Ethics:</strong> Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PO9</td>
<td><strong>Individual and team work:</strong> Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.</td>
</tr>
<tr>
<td>PO10</td>
<td><strong>Communication:</strong> Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.</td>
</tr>
<tr>
<td>PO11</td>
<td><strong>Project management and finance:</strong> Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.</td>
</tr>
<tr>
<td>PO12</td>
<td><strong>Life-long learning:</strong> Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.</td>
</tr>
<tr>
<td>PSO</td>
<td><strong>PSO Statement</strong></td>
</tr>
<tr>
<td>PSO1</td>
<td>Apply information science theory, algorithmic and programming principles for comprehending technological trade-off in order to determine conceptual aspects of real world problems in information technology.</td>
</tr>
<tr>
<td>PSO2</td>
<td>Analyze and create problem frames in order to formulate decomposition structure of information technology problem with correct resources, infrastructure and technology requirements determination for solution realization.</td>
</tr>
<tr>
<td>PSO3</td>
<td>Compose technical design specifications using template based approaches for formally expressing the solution implementation by applying techniques and methods to create, enhance, and deliver IT tools with appropriate CASE tools selection.</td>
</tr>
<tr>
<td>PSO4</td>
<td>Exercise research and development aptitude focusing knowledge creation and dissemination through engineering artifacts construction, preparation and presentation of engineering evidences using procedures, techniques, guidelines, and standards considering technology migration and evolution.</td>
</tr>
</tbody>
</table>
Assessment Pattern for all engineering courses with lab

<table>
<thead>
<tr>
<th>In Semester Assessment</th>
<th>End Semester Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment (%)</td>
<td>Lab Assessment (%)</td>
</tr>
<tr>
<td>10</td>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hands on</th>
<th>Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viva/Lab Exam (%)</td>
<td>Lab Assessment (%)</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

Assessment pattern for Mathematics 3, Mathematics 4 - direct grade Entry.

<table>
<thead>
<tr>
<th>In Semester Assessment</th>
<th>VIVA</th>
<th>End Semester Exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment (%)</td>
<td>VIVA (%)</td>
<td>End Semester Exam (%)</td>
</tr>
<tr>
<td>50</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>Mid Semester Exam</td>
<td>Assignment</td>
<td>30</td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Subject No.</td>
<td>Subject Code</td>
<td>Subject Name</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>S1</td>
<td>IT4085</td>
<td>Distributed Computing</td>
</tr>
<tr>
<td></td>
<td>IT4018</td>
<td>Convergence Technology</td>
</tr>
<tr>
<td></td>
<td>IT4006</td>
<td>Artificial intelligence</td>
</tr>
<tr>
<td></td>
<td>IT4190</td>
<td>Financial Technology</td>
</tr>
<tr>
<td>S2</td>
<td>IT4012</td>
<td>Image Processing</td>
</tr>
<tr>
<td></td>
<td>IT4019</td>
<td>Software Design Methodology</td>
</tr>
<tr>
<td></td>
<td>IT4115</td>
<td>Major Project</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subject No.</th>
<th>Branch</th>
<th>Information Technology</th>
<th>Year: SY</th>
<th>Academic Year: 2019-20</th>
<th>B19</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT4117</td>
<td></td>
<td>Subject Code: IT4117</td>
<td>Subject Name: Industry Internship</td>
<td>Teaching Scheme</td>
<td>Examination Scheme</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Credit: 16</td>
<td></td>
</tr>
<tr>
<td>IT4032</td>
<td></td>
<td>Subject Code: IT4032</td>
<td>Subject Name: International Internship</td>
<td>Teaching Scheme</td>
<td>Examination Scheme</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Credit: 16</td>
<td></td>
</tr>
<tr>
<td>IT4030</td>
<td></td>
<td>Subject Code: IT4030</td>
<td>Subject Name: Research Assistance</td>
<td>Teaching Scheme</td>
<td>Examination Scheme</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Credit: 16</td>
<td></td>
</tr>
<tr>
<td>IT4031</td>
<td></td>
<td>Subject Code: IT4031</td>
<td>Subject Name: Project Internship</td>
<td>Teaching Scheme</td>
<td>Examination Scheme</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Credit: 16</td>
<td></td>
</tr>
</tbody>
</table>
IT4085: Distributed Computing

Credits: 04
Teaching Scheme: Theory: 3 Hours / Week
Lab: 2 Hours/Week

Section 1:
Clock Synchronization, Logical Clocks, Scalar time, Vector time, Global State, Election Algorithm: Bully Algorithm, Ring Algorithm, Mutual Exclusion: Requirements, Performance metrics, Centralized algorithm, Lamport’s algorithm, Distributed algorithm, Token Ring algorithm.

Section 2:
Case study: Google File System

List of Practicals (Any 6 statements):

1. Write a program to implement Client-Server application using RMI using RMI for remote computation.
2. Write a program to implement Client-Server application using RPC mechanism for remote computation.
3. Write a program to simulate Cristian's / Berkeley ’s clock synchronization algorithm.
4. Write a program to simulate Lamport’s clock synchronization algorithm.
5. Write a program to simulate Ring/Bully Election algorithm.
6. Write a program for implementation of Deadlock through simulation.
7. Write a program to simulate the Centralized/Ricart Agrawala’s / Lamport algorithm for mutual exclusion.
8. Write a program to simulate Wait for Graph based Centralized algorithm for deadlock detection.

9. Write a program to simulate Byzantine Generals problem.

10. Design a distributed application which consists of a server and client using threads.

11. Study and Configure Hadoop.

12. Design a distributed application using MapReduce under Hadoop for Character counting in a given text file and Counting no. of occurrences of every word in a given text file.

List of Project areas (Any 1 project):

1. Implement concurrent client-server application.

2. Design and develop a distributed Hotel booking application using Java RMI.
   
   A distributed hotel booking system consists of the hotel server and the client machines.
   
   The server manages hotel rooms booking information.
   
   A customer can invoke the following operations at his machine i) Book the room for the specific guest ii) Cancel the booking of a guest iii) Enquire the check in date for the specified customer/guest.

3. Design a distributed application using Mapreduce to Analyze the Million Song dataset.

4. Design a distributed application using MapReduce which processes a log file of a system.
   List out the users who have logged for maximum period on the system. Use simple log file from the Internet and process it using a pseudo distribution mode on Hadoop platform.

Text Books:


Reference Books:


Course Outcomes:

The student will be able to –

1. Identify the basic principles, design issues and architectural aspects of distributed systems. (2)

2. Analyze the different techniques used for Communication in distributed system. (2)

3. Develop the solutions for Clock synchronization, Mutual exclusion in distributed system. (3)

4. Construct an optimal and cost-effective solution for Distributed transaction and Deadlock. (4)

5. Use and apply important methods in distributed systems to support Scalability and Fault tolerance. (3)

6. Gain knowledge on Distributed File System and design issues of Distributed Shared Memory. (3)
IT4018: Convergence Technology

Credits: 4

Teaching Scheme: Theory: 3 Hours / Week
Lab: 2 Hours/Week

Section I:
Analog telephony - Basic Circuits (loopstart, Groundstart), Lines and Trunks, DTMF, CP Tones, Supervisions, Disadvantages of Analog telephony, Basic Call Fence post diagram. Digital Telephony - Advantages, Signaling Mode (CAS, CCS) Digital Telephony: T1/E1 Circuits, Clock Synchronization, Line coding, Framing formats, Standards and Standardization bodies, Basic ISDN Protocol Stack, ISDN Reference points, Bearer and D-Channel. Digital Telephony: Q.931 Protocol, Basic ISDN Signaling fence Post diagram, Nodal Messages, Timers, Information Elements, Code Points, Code Sets Supplementary Services, Integrated Voice, Video and Data, Cost of basic and supplementary services, Requirements of an Enterprise - Integration of e-mail, chat, mobility, user identity, video/content sharing, Motivation for VoIP, Integration of Voice and Data networks.


Section II:
Business Use cases: Enterprise and Service providers (AKA Carriers), Typical enterprise Topology, Types of enterprises and business needs. Basic features: Hold, Transfer, Conference, Forward, Coverage/Voicemail, Bridging, Mobility; Collaboration: Integration of Voice, Video, Content and enterprise communication channels such as email, IM, web applications. Use cases and call flows. Introduction to Contact Center, Contact Center types, Contact Center Terminologies, Contact Center Infrastructure and Technology, Activity/Quiz – I, Contact Center Use Cases and applications, Q&A, Activity/Quiz – II, Voice application protocols - VXML, CCXML, MRCP, Outbound Contact Center Application flows - voice, SMS, email, custom solutions; Activity/Quiz – III, CTI Application protocols - CSTA, TSAPI, JTAPI, Inbound Contact Center Application flows - voice, email, chat, social media, custom solutions; Activity – IV, Contact Center Reporting, Recording, WFM, WFO, Q&A, Activity/Quiz – II, Cloud Computing: UCaaS, CCaaS, Client SDKs and integration of VoIP with phone apps. Use cases such as Bank applications, Retail online shops. Components for VoIP enablement supported in Android, Components for VoIP enablement supported in IOS, WebRTC, Analytics in Voice & Data, Buffer.
NOTE: This course will be conducted by industry faculty and hence assignment list and project will be decided by industry person.

**Text Books:**

**Reference Books:**

**Additional Reading:**

**Course Outcomes:**
Upon completion of the course, graduates will be able to -
1. Categorize voice and data networks based on various protocols.
2. Analyze the protocols and standards for converged networks.
4. Design the converged network to fulfill the societal requirement.
5. Judge the impact and benefits of converged network in exploitation on environment and society.
6. Prepare cost effective solutions to fulfill the need of convergence technology.
IT4006: Artificial Intelligence

Credits: 04

Teaching Scheme:-Theory: 3 Hours / Week
Lab: 2 Hours/Week

Section 1: Topics/Contents
Fundamentals of Artificial Intelligence

Uninformed Search Strategies
Formulation of real world problems, Breadth First Search, Depth First Search, Depth Limited Search, Iterative Deepening Depth First Search, Bidirectional Search, Comparison of Uninformed search Strategies, Searching with partial information, Sensor-less problems, Contingency problems

Informed Search Strategies
Generate & test, Hill Climbing, Best First Search, A* and AO* Algorithm, Constraint satisfaction, Game playing: Minimax Search, Alpha-Beta Cutoffs, Waiting for Quiescence, Applications of search strategies:- Tic-tac-Toe, 8-Puzzle

Section2: Topics/Contents
Knowledge Representation
Knowledge based agents, Wumpus world. Propositional Logic: Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining.
Expert System: Case study of Expert System in PROLOG

Introduction to Planning and ANN

Uncertainty and Expert system
Introduction to Expert Systems, Architecture of Expert Systems; Expert System Shells; Knowledge Acquisition; Case Studies; MYCIN, Learning, Rote Learning; Learning by Induction; explanation based learning.
List of Practical’s
Complete any six lab assignments including either 10 or 11.
1. Implement Non-AI and AI Techniques
2. Implement simple hill climbing for 8-puzzle/other application
3. Implement steepest ascent hill climbing for 8-puzzle/other application
4. Implement Best First Search & A* algorithm for 8-puzzle/other application
5. Implement Perceptron learning algorithm for 2 class classification problem
6. Implement real time applications in Prolog.
7. Expert System in Prolog- new application
8. Implement any two Player game using min-max search algorithm.
9. Design a fuzzy set for shape matching of handwritten character
10. Apply c-means clustering for pattern recognition in your domain of interest
11. Apply k-NN classifier for pattern recognition in your domain of interest
12. Write a program to extract statistical features from an image of a hand written digit.

List of Project areas
Following is the indicative list. Projects are not limited to only given list. Teacher and student can jointly decide the project area other than not listed here.

1. Medical diagnosis- Imaging and non-imaging approaches
2. Visual pattern clustering/Pattern clustering
3. Pattern classification
4. Neural networks as classifiers
5. Neural networks for pattern clustering
6. 2D/3D-Object recognition/detection
8. Speech analysis/processing/Recognition
9. Natural language processing/Understanding
10. AI for cyber security- Palm print, Finger print and thumb print and other approaches
11. Robotic control
12. AI in agricultural- crop and soil monitoring and etc

Text Books
Reference Books
2. Eugene, Charniak, Drew Mcdermott: "Introduction to Artificial Intelligence.", Addison Wesley
5. Jacek M. Zurada, Introduction to artificial neural systems, Jaico Publication
6. Dan W. Patterson, “Artificial Intelligence and Expert systems”, PHI publication

Course Outcomes:
Upon completion of the course, graduates will be able to -
1. Understand the basics of the theory and practice of Artificial Intelligence as a discipline and about intelligent agents capable of problem formulation.
2. Evaluation of different uninformed search algorithms on well formulated problems along with stating valid conclusions that the evaluation supports.
3. Design and Analysis of informed search algorithms on well formulated problems.
4. Formulate and solve given problem using Propositional and First order logic.
5. Apply planning and neural network learning for solving AI problems
6. Design a small expert system using PROLOG.
IT4190: Financial Technology

Credits: 04

Teaching Scheme:- Theory: 3 Hours / Week
Lab: 2 Hours/Week

Unit I - Introduction to Banking, Finance and Banking Regulations
Introduction to different kind of Banking, Banking Tiers (Front office, Middle office and Back office), Introduction to primary and secondary Markets, Mergers and Acquisitions, Market Participants, Introduction to Regulatory world (BASEL III, Mifid, FATCA etc.)

Unit II - Macro Economics
Introduction to the macroeconomics environment and policy players, Gross Domestic Product (GDP), Introduction to the Fiscal policy and Monetary policy, Introduction to Fiscal deficit, Current account deficit, repo rate, reverse repo rate, Introduction to types of Money, M0, M1, M2, M3 … etc. Keynesian macroeconomics: Backdrop of the Great Depression and fall of classicism, Keynesian Multiplier: Consumption function and Investment function,

Unit III - Basics of Corporate Finance
Portfolio Theory, Capital Asset Pricing Model – CAPM, Cost of Capital, NPV & IRR

Unit IV - Equity and Fixed Income
Basics of Risks and Returns, Introduction of Equity Instruments and their history, Introduction of Debt Instruments (Bonds) and their history, Equity Valuations, Bond Valuations

Unit V – Derivatives
Introduction to Derivatives, Understanding Forwards, Futures and Options, Introductions to Swaps, Derivative pricing models: Black-Scholes, Binomial, Options Strategy and Options Greeks

List of Practices:
1. Trade Lifecycle
2. Explanation of Lab Work
3. Agile Methodology
4. Test Driven Development on unit 3

Text Books & Reference Books:
IT4012: Image Processing

Credits: 4  
Teaching Scheme: 3 Hours / Week  
Lab: 2 Hours / Week

Section 1:
Introduction, Elements of image processing system, Scenes and Images, Vector Algebra, Human Visual System, color vision color model: RGB, HVS, YUV, CMYK, YC\_bC\_r and some basic relationships between pixels, linear and nonlinear operations. Image types (optical and microwave), Image file formats (BMP, tiff, jpeg, ico, ceos, GIF, png, raster image format). Image sampling and quantization. Thresholding, Spatial domain techniques

- Image Negative, Contrast stretching, gray level slicing, bit plane slicing, histogram and histogram equalization, local enhancement technique, image subtraction and image average,

Image segmentation- Classification of image segmentation techniques: Watershed Segmentation, Edge-based Segmentation, region approach, clustering techniques, edge-based, classification of edges and edge detection, watershed transformation.

Section 2:
Introduction to Image compression and its need, Coding redundancy, classification of compression techniques (Lossy and lossless- JPEG, RLE, Huffman, Shannon fano), Objective Recognition { Need, Automated object recognition system, pattern and pattern class, relationship between image processing and object recognition, approaches to object recognition.


I: List of Practical (Any 6)

1. Write matlab code to display following binary images
   a. Square
   b. Triangle
   c. Circle
   d. Write matlab code to perform following operations on images
   e. Flip Image along horizontal and vertical direction.
   f. Enhance quality of a given image by changing brightness of image.
   g. Image negation operation.
   h. Change contrast of a given Image.

2. Write Matlab code to implement pseudo coloring operation of a given image. Write Matlab Code for Pseudo Colour of Image by using Gray to colour transform.
3. Study of different file formats e.g. BMP, TIFF and extraction of attributes of BMP.

4. Write matlab code to find following statistical properties of an image.
   a. Mean
   b. Median
   c. Variance
   d. Standard deviation
   e. Covariance.

5. Write matlab code to enhance image quality by using following techniques
   a. Logarithmic transformation
   b. Histogram Equalization
   c. Gray level slicing with and without background.
   d. Inverse transformation.

6. Read an Image and Perform singular value decomposition. Retain only k largest singular values and reconstruct the image. Also Compute the Compression ratio.

7. Write matlab code to enhance image quality by using following techniques
   a. Low pass and weighted low pass filter.
   b. Median filter.
   c. Laplacian mask.

8. Write matlab code for edge detection using Sobel, Prewitt and Roberts operators.

9. Write C-language code to find out Huffman code for the following word COMMITTEE.

10. Write matlab code to design encoder and decoder by using Arithmetic coding for the following word MUMMY. (Probabilities of symbols M-0.4, U-0.2, X-0.3, Y- ).

11. Write matlab code to find out Fourier spectrum, phase angle and power spectrum of binary image and gray scale image.

II: Projects: (Any 1)
1. Pseudo colour image processing
2. Image Editing
3. Video Editing
4. Image Compression
5. Video Compression

Text Books:
Reference Books:

Course Outcomes:
The student will be able to
1. Describe image model
2. Perform spatial filtering on image
3. Identify Image Segmentation techniques.
4. Apply lossless and lossy compression techniques for image compression.
5. Use various image transforms to analyze and modify image.
Section 1:


**System Behaviour Specification**: Static behaviour: Use case, Use case digram components, use case diagram, Actor Generalization, Include and Extend, Template for use case Narrative, Building Domain model, and capturing system behaviour in use cases.

Dynamic Behaviour: Sequence diagram, Object lifelines, and message types, modelling collections multiobject, refining sequence diagram, collaboration diagram, states, events and actions, Nested machines and concurrency, Modifying the object model to facilitate states, Modeling methods with activity diagram, Activity Diagrams: Decisions and merges, synchronization, Iteration, partitions, parameters and pins, Expantion region, swimlanes, concurrency and synchronization, Communication diagram, Interaction overview diagrams, Timing Diagrams.


Section 2:

**System Design Specification**: Design of Software Objects, Features and Methods, Cohesion and Coupling between Objects, Coupling and Visibility, Interfaces, Interfaces with Ball and Socket Notation, Templates, Analysis model vs. design model classes, Categorizing classes: entity, boundary and control, Modeling associations and collections, Preserving referential integrity, Achieving reusability, Reuse through delegation, Identifying and using service packages, Improving reuse with design. Packages and interfaces: Distinguishing between classes/interfaces, Exposing class and package interfaces, Subscribing to interfaces Component and deployment diagrams: Describing dependencies, Deploying components across threads, processes and processors.

**Design Patterns**: Introduction to Design Pattern, Describing Design Patterns, Catalogue of Design Patterns Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype,


**List of Practicals: (For THL, TLP courses)**

1. To narrate Requirement Definition Document for the target system with following three areas: Problem Identification, Problem Definition, and Problem Statement
3. To create Business Process Diagrams for all the scenarios identified using BPMN 2.0 and BPM practices. Process modeling captures the ordered sequence of activities within a process along with supporting information from process modeling, the business process is framed in a BPD to reflect the activities, the roles that conduct those activities, conditional branching, and the sequence of the workflow between the activities.
4. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system. Specify the behavior system and of the target map requirements to Use cases.
   a. The System context Diagram depicts the overall System behavioral trace and Requirement Capture diagram depicts the hierarchy Organization. The Use al Use case Case diagram should encompass
      b. Actors (External Users)
      c. Transactions (Use Cases)
      d. Event responses related to transactions with external agents.
      e. Detection of System boundaries indicating scope of system.
5. To depict the dynamic behavior of the target system using sequence diagram. The Sequence diagram should be based on the Scenarios generated by the inter-object Communication. The model should depict:
   a. Discrete, distinguishable entities (class).
   b. Events (Individual stimulus from one object to another).
   c. Conditional events and relationship representation.
6. To depict the state transition with the life history of objects of a given class model. The model should depict:
a. Possible ways the object can respond to events from other objects.
b. Determine of start, end, and transition states.

7. To depict the dynamic behavior using detailed Activity diagram. Activity is a parameterized behavior represented as coordinated flow of actions. The flow of execution is modeled as activity nodes connected by activity edges.
   - A node can be the execution of a subordinate behavior, such as an arithmetic computation, a call to an operation, or manipulation of object contents.
   - Activities may form invocation hierarchies invoking other activities, ultimately resolving to individual actions.

8. To develop logical static structure of target system with Software Class diagram. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from System analysis phase. The design model should depict
   a. Relationship between classes: inheritance, Aggregation, Instantiation Assertion,
   b. Identification of objects and their purpose.
   c. Roles / responsibilities entities that determine system behavior.

9. To enhance Software Class diagram to Architecture diagram with appropriate design patterns. The patterns selected shall be justifiable and applied to individual and distinct hierarchies. Suitable Architectural Styles shall be selected and the elements shall be well-documented.

10. To represent physical module that provides occurrence of classes or other logical elements identified during analysis and design of system using Component diagram. The model should depict allocation of classes to modules. To narrate precise Program Design Language constructs separating computation from interface. To represent deployment view of the system through Diagram. Architecture

Text Books:

Reference Books:


Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Examine and breakdown real-world problem scenarios into structured partitions depicting static and dynamic behavior of the system using business process management practices, object-oriented analysis principles and Model Driven Development practices.

2. Identify and formulate software requirements and behavioral models using static and dynamic behavioral views indicating structured problem partitioning and state-based exploration.

3. Compose system analysis and design specifications indicating logical, physical, deployment, and concurrency viewpoints using object-oriented analysis and design principles and Model Driven Engineering practices.

4. Construct and justify the evolutionary system description models expressing high-level architecture accommodating applicable architectural styles compatible to requirements and behavioral models using UML-supported modeling tools.

5. Comprehend the nature of design patterns by understanding a small number of examples from different pattern categories and apply these patterns in creating a correct design using design heuristics, published guidance, applicability, reasonableness, and relation to other design criteria resulting in well-documented system profiles to the engineering and social community.

6. Propose multi-faceted defendable solutions demonstrating accommodating design patterns reducing the potential cost performance and impedance in order to realize system artifacts with the help of Model Driven
IT4075: Design and Analysis of Algorithms

Credits: 4

Teaching Scheme:- Theory: 3 Hours / Week
Lab: 2 Hours / Week

Section 1:
Basic introduction, time complexity analysis, Divide and Conquer-Asymptotic notations (Big Oh, small oh, Big Omega, Theta notations). Best case, average case, and worst case time and space complexity of algorithms. Overview of searching, sorting algorithms. Adversary lower bounds (for comparison based sorting, for finding second minima). Using Recurrence relations and Mathematical Induction to get asymptotic bounds on time complexity. Master’s theorem and applications. Proving correctness of algorithms.
Divide and Conquer- Analyzing Quick sort, Randomized Quick sort, Merge sort, Counting Inversions, Finding majority element, Finding Median, Efficient algorithms for Integer arithmetic (Euclid's algorithm, Karatsuba's algorithm for integer multiplication, fast exponentiation), Finding closest pair of points in plane, computing convex hull of points in plane, basic idea of FFT algorithm and applications
Dynamic Programming- General strategy, simple dynamic programming based algorithms to compute Fibonacci numbers, binomial coefficients, Matrix Chain multiplication, Optimal binary search tree (OBST) construction, Coin change problem, 0-1 Knapsack, Traveling Salesperson Problem, All pair shortest path algorithm, Longest increasing subsequence problem, Longest common subsequence problem, Largest independent set for trees.
Greedy- Analysis and correctness proof of minimum spanning tree and shortest path algorithms, Huffman coding, conflict free scheduling, fractional knapsack.

Section 2:
Linear Programming: Introduction to linear programming, geometric interpretation, LP duality, Simplex algorithm, Linear optimization problems and their LP formulation.
Flows and Matchings- Flows: Flows in the network, Max-flow min-cut theorem, Ford Fulkerson's algorithm, LP formulation of flow problem, Applications (e.g. image segmentation, airline scheduling)
Matchings: Perfect matchings in bipartite graphs, LP formulation, Hall's marriage theorem, König's theorem, augmenting path algorithm for matchings.
Introduction to Approximation algorithms, NP-optimization problems, Approximation algorithm for Vertex Cover, Traveling Sales Person Problem(TSP), Set-cover.
I. List of Practicals:

1. Quick Sort implementation using divide and conquer approach. Time complexity measure is to be obtained.
2. Merge Sort implementations using divide and conquer approach.
3. Job scheduling using Greedy approach
4. Minimal spanning Trees using Greedy approach
5. Finding shortest path for multistage graph problem. (single source shortest path and all pairs shortest path.)
6. 0/1 knapsack problem using Greedy approach.
7. n-Queen problem using Greedy approach.
8. Traveling salesman problem using Greedy approach.
9. 0/1 knapsack problem using Dynamic Programming.
10. n-Queen problem using general backtracking method.
12. Find a subset of a given set \( S = \{s_1, s_2, \ldots, s_n\} \) of \( n \) positive integers, whose sum is equal to a given positive integer \( d \).

II. List of Projects:

1. Finding shortest path to reach from one city to other.
2. Map system for travelling through roads.
3. Huffman coding and decoding
4. Shared memory management
5. Task allocation to employee
6. Plagiarism detection
7. Packet routing
8. Puzzle/Maze game
9. Geographical search
10. Pattern detection
11. Clustering patterns
12. Classification of unstructured data

Text Books: (As per IEEE format)


Reference Books: (As per IEEE format)

Course Outcomes:
The student will be able to –
1. Formulate computational problems in abstract and mathematically precise manner.(1)
2. Design efficient algorithms for computational problems using appropriate algorithmic paradigm.(2)
3. Analyze asymptotic complexity of the algorithm for a complex computational problem using suitable mathematical techniques.(3)
4. Formulate computational problem as linear program and apply LP, network flow, based techniques to design efficient algorithms for them.(4)
5. Establish Incompleteness of some decision problems, grasp the significance of the notion of Incompleteness and its relation with intractability of the decision problems and design efficient approximation algorithms for standard NP-optimization problems.(5)
6. Incorporate appropriate data structures, algorithmic paradigms to craft innovative scientific solution for complex computing problems.(4)
Section 1:
Descriptive Statistics
Mechanisms of data collection and challenges involved therein. Typical preprocessing operations: combining values into one, handling incomplete or incorrect data, handling missing values, recoding values, sub-setting, sorting, transforming scale, determining percentiles, data manipulation, removing noise, removing inconsistencies, transformations, standardizing, normalizing - min-max normalization, zscore standardization, and rules of standardizing data, role of statistics in analytics, types of data (scales of measurement-NOIR), data distributions, measures of variability (range, quartile, five number summary, variance, std dev, coeff of variation), analyzing distributions,Chebychev’s inequality, measures of shape (skewness, kurtosis), measures of association (covariance, correlation), outliers

Inferential Analytics
Role of probability in analytics. Need for sampling, generating samples, sampling and non-sampling error. Sampling Distribution of Mean, Central Limit Theorem, Standard Error. Estimation: Point and Interval Estimates, Confidence Intervals, level of confidence, sample size.

Hypothesis Testing: basic concepts, Errors in hypothesis testing, Power of test, Level of significance, p-value, general procedure for hypothesis testing. Parametric tests – z test, t test, chi-square test. Hypothesis testing of means: two tailed and one-tailed tests. Chi square test for independence and goodness of fit. Hypothesis testing for comparing two related samples. Limitations of hypothesis testing. Picking up the right test for a given scenario.

Predictive Analytics: Regression
Correlation and regression, Simple Linear Regression Model, Least Squares Method. Making Data Models more flexible, making data models more selective, dealing with Categorical variables, Interpretation of regression coefficients, fine tuning data models (assessing the fit, model fitting), Coefficient of determination, Significance tests, Residual analysis, Prediction intervals. Model evaluation techniques. Assumptions of regression analysis.

Section 2:
Predictive Analytics: Supervised Method
Bias-Variance Dichotomy, Model Validation Approaches, Logistic Regression Linear Discriminate Analysis, Quadratic Discriminate Analysis, Regression and Classification Trees, Support Vector Machines, Ensemble Methods: Random Forest Neural Networks, deep learning.

Predictive Analytics: Unsupervised Method
Similarity Measures, Design of recommender systems, user based and item based Collaborative filtering, Clustering, Associative Rule Mining
Prescriptive Analytics
Creating data for analytics through designed experiments, creating data for analytics through Active learning, creating data for analytics through Reinforcement learning

List of Practicals:
1. Getting started with R
2. Using R for data preprocessing
3. Performing exploratory analysis for any dataset
4. Data visualization using ggplot library
5. Performing correlation and regression analysis
6. Implementing hypothesis testing and chi square test
7. Data analysis case study using R for a readily available data set
8. Predict the income class of US population from Census Income Dataset.
9. Recommend new movies to users using Movie Lens Dataset.
10. Identify the tweets which are hate tweets and which are not using Twitter Classification Dataset
11. Implementing a simple Recommender System based on user buying pattern.
12. Data analysis case study using R for readily available data set using any one machine learning algorithm

List of Project areas:
1. Predict the class of the flower based on available attributes on IRIS data set.
2. Predict if a loan will get approved or not using Loan Approval dataset
3. Predict the sales of a store using Big Mart sales dataset
4. Predict the median value of owner occupied homes using boston housing data set
5. Use classification and clustering techniques to deal with the data using Turkiye Student Evaluation Dataset.
6. Predict the activity category of a human using Human Activity Recognition Dataset.
7. Classify the documents according to their labels using Text Mining dataset.
8. Predict the income class of US population from Census Income Dataset.
9. Recommend new movies to users using Movie Lens Dataset.
10. Identify the tweets which are hate tweets and which are not using Twitter Classification Dataset
11. Implementing a simple Recommender System based on user buying pattern.

Data analysis case study using R for readily available data set using any one machine learning algorithm

Text Books:
1. Business Analytics by James R Evans, Pearson

Reference Books:
Course Outcomes:
The student will be able to:
1. Understand the process of converting data into a required format required for particular analysis.(2)
2. Analyze data, test claims, and draw valid conclusions using appropriate statistical methodology.(2)
3. Utilize statistical tools in deriving insights from data.(3)
4. Apply analytic techniques and algorithms (including statistical and data mining approaches) to large data sets to extract meaningful insights.(3)
5. Use appropriate resources to research, develop and contribute to advances and trends within the field of Data Science.(5)
6. Interpret and present visually, orally and in written form, valid conclusions drawn from data analysis.(5)
IT4077: Artificial Neural Networks

Credits: 04

Teaching Scheme:
- Theory: 3 Hours / Week
- Lab: 2 Hours/Week

Section 1: Topics/Contents

Introduction to ANN

History of Neural networks, Motivation, Introduction to Neural networks: basics, comparison of human brain and machine, biological neuron, general neuron model, activation functions, applications and advantages of neural networks, McCulloch-Pitts model, Neural net architecture, Neural learning

Introduction to ANN

Neural network learning in general, concept of local and global minima, general learning rule, introduction to various learning rules like, perceptron, Widro-Hoff, Winner-takes-all and dela-Learning rule

Supervised Learning


Section 2: Topics/Contents

Unsupervised & Associative Learning


Associative Learning

Associative non iterative procedures for association, Hop field networks, Optimization, Learning using Hopfield networks, Brain state in a box network, Boltzman machines, Hetero-associators.

Fuzzy -neural hybrid systems

Introduction to fuzzy sets: definition, types, why fuzzy sets, fuzzy membership function, properties of fuzzy sets.
Study of one complete hybrid system of fuzzy neural networks for real world pattern recognition

List of Practical’s

Complete any six lab assignments including either 10 or 11.

1. Implementation of single perceptron rule
2. Implementation of delta-learning rule using single neuron
3. Implementation of Widro-Hoff learning rule
4. Implementation of single layer perceptron for R-category patterns
5. Use of various activation functions in perceptron
6. Implement a back propagation algorithm
7. Design a fuzzy set for measuring similarity of features of two handwritten digits
8. Use fuzzy-perceptron as hybrid model
9. Implement a max net for real world problem
10. Implement a competitive net for real world problem
11. Implement a Hopfield network for real world problem
12. Use single perceptron to classify 2-class problem using data sets available on UCI repository.

**List of Project areas**
Following is the indicative list. Projects are not limited to only given list. Teacher and student can jointly decide the project area other than not listed here.

1. Medical diagnosis using ANN- Imaging and non-imaging approaches
2. Visual pattern clustering/Pattern clustering
3. ANN for prediction
4. Neural networks as classifiers (single/multiple layers)
5. Neural networks for pattern clustering (single/multiple layers)
6. ANN for 2D/3D-Object recognition/detection
7. Fuzzy neural as a hybrid system/Deep Learning approaches for complex visual pattern recognition
8. ANN for Speech analysis/processing/Recognition
9. Natural language processing/Understanding
10. AI for cyber security- Palm print, Finger print and thumb print and other approaches
11. ANN for Robotic control
12. ANN in agricultural applications- crop and soil monitoring and etc

**Text Books**
1. Jacek M Zurada, “Introduction to Artificial Neural systems”, Jaico Publication
2. Elements of Artificial Neural Networks - by Kishan Malhotra, Chilukurik. Mohan, Sanjay Ranka Penram International Publishing (India) Pvt. Ltd. Second edition,
3. Timothy J. Ross, “Fuzzy logic with Engineering applications”,

**Reference Books**
1. Neural Network and Fuzzy system by Bart Kosko, John c. Burgess.
2. Fundamental of Artificial Neural Networks. By M.H. Hassoun.
4. Relevant Research Papers suggested by the teacher

**Course Outcomes:**
Upon completion of the course, graduates will be able to -
The student will be able to –
1. Apply neural network architecture for solving real world applications
2. Solve problems using supervised learning techniques.
4. Solve problems using hybrid-fuzzy neural system
5. Apply theory of fuzzy sets for complex pattern recognition applications
6. Decide the neural architecture, hidden layers and the number of neurons used in output layer.