**DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**B.TECH(IT) SYLLABUS (2018 ADMITTED)**

**THIRD SEMESTER**

**MAT 2155: ENGINEERING MATHEMATICS - III [2 1 0 3]**

Boolean Algebra: Partial ordering relations, Poset, Lattices, Basic Properties of Lattices. Distributive and complemented lattices, Boolean lattices and Boolean Algebra. Propositional and Predicate Calculus: Well-formed formula, connectives, quantifications, Inference theory of propositional and predicate calculus. Elementary configuration: Permutations and Combinations, Generating function, Principle of inclusion and exclusion Partitions, compositions. Ordering of permutations: Lexicographical and Fikes. Graph theory: Basic definitions, Degree, regular graphs, Eulerian and Hamiltonian graphs, Trees and Properties, Center, radius and diameter of a graph, Rooted and binary trees, Matrices associated with graphs, Algorithms for finding shortest path, Algorithm. Group theory: Semi groups, Monoids, Groups- subgroups, Normal Subgroups, Cosets, Lagrange’s Theorem, Cyclic groups.

**References:**

1. Liu C.L., Elements of Discrete Mathematics (2e), McGraw Hill, New Delhi, 2007.
2. Trembaly J.P. and Manohar R.., Discrete Mathematics Structures with application to computer Science, Tata McGraw Hill, 2012.
3. Page E.S. and Wilson L.B., An Introduction to Computational Combinatorics, Cambridge Univ. Press, 1979.
4. Narasingh Deo, Graph theory with Applications to computer science, PHI, 2012.

**ICT 2153: DATA STRUCTURES [3 1 0 4]**

**Objectives:**

* To summarize various sorting and searching techniques
* To identify the appropriate data structure for a specific application.
* To apply data structure concepts for efficient representation of data

**Abstract:**

Introduction, Programming fundamentals, Stacks, Queues and their applications, Sparse Matrix, Pointers and dynamic memory allocation, Linked Lists: Singly linked lists, Dynamically Linked Stacks and Queues, Polynomial representation and polynomial operations using singly linked list, Singly Circular Linked List, Doubly Linked Lists, Trees: Binary trees, Heaps, Binary Search Trees, Threaded binary trees, Graphs: Depth First Search, Breadth First Search, Connected components, Spanning trees, Sorting and searching Techniques.

**Syllabus:**

**Introduction:**

Performance Analysis and Measurements, Introduction to data structure, Arrays, Strings, Functions, Bubble sort, Insertion sort, Selection sort, Linear and Binary search, Objects and Classes, Abstract data types (ADT) **[6 Hours]**

**Stacks:**

Definition, Operations on stack, Evaluation of Arithmetic Expressions, Conversion of arithmetic expressions, Recursion, Multiple Stacks **[5 Hours]**

**Queues:**

Definition, Operations on queue and Circular queue, Applications. **[2 Hours]**

**Sparse Matrix:**

Representations and Transpose techniques **[3 Hours]**

**Linked Lists:**

Introduction to pointers and Dynamic memory allocation, Singly linked lists, Circular lists, Dynamically Linked Stacks and Queues, Polynomial representation and polynomial operations using singly linked list, Singly circular linked list, Doubly linked lists. **[12 Hours]**

**Trees:**

Tree terminology, Binary trees, Strictly binary tree, Complete binary tree, Memory representation of binary tree, Abstract Data Type, Properties, Binary tree representations, Binary Tree Traversal algorithms, Expression tree, Threaded binary tree, Decision tree, Copying and testing equality, Binary tree applications, Heaps, Binary Search Trees. **[12 Hours]**

**Graphs:**

Definitions and Representations, Depth First Search, Breadth First Search, Connected components, Spanning trees. **[3 Hours]**

**Sorting Techniques:**

Quick Sort, Merge sort, Heap sort, Radix sort **[5 Hours]**

**Outcome:**

By the end of this course, the students are able to:

* Relate the concepts of arrays, dynamic memory management, class, searching, sorting.
* Illustrate the working of linear and non-linear data structure.
* Apply the appropriate data structure to solve real world problems

**References:**

1. Ellis Horowitz, Sartaj Sahni, Dinesh Mehta, *Fundamentals of Data Structures in C++* (2e)*,* Galgotia Publications, 2008.
2. Mark Allen Weiss, *Data Structures and Algorithm Analysis in C++* (3e), Pearson Education, 2009.
3. Michael T. Goodrich, Roberto Tamassia, David Mount, *Data Structures and Algorithms in C++* (2e), John Wiley & Sons, 2011.
4. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed*, Fundamentals of Data structures in C* (2e)*,* Silicon Press, 2008

## **ICT 2154: DIGITAL SYSTEMS [3 1 0 4]**

**Objectives:**

* To explain the elements of digital system abstractions such as Boolean algebra, flip-flops, finite-state machines (FSMs) and memory devices.
* To illustrate simplification of Boolean expressions using Boolean theorems, K-Map and tabulation methods.
* To demonstrate combinational and sequential logic circuits design.

**Abstract:**

Introduction, Simplification of Boolean functions – K-map and tabulation method, NAND and NOR implementation, Combinational logic- Design of Adders/Subtractors, Binary Parallel adder[7483], Carry Look ahead Adder [74182], Multiplier using 7483, BCD adder, Magnitude Comparator [7485], Decoder [74138,7442], Combinational logic circuit design using decoders, Encoder [74148], Multiplexers [74157, 74153], Combinational logic circuit design using multiplexers, De Multiplexers, ROMS and Programmable Logic Arrays, Sequential logic –Asynchronous and Synchronous counters, Synchronous counter design, Shift registers, Shift register counters, Analysis and design of clocked sequential circuits, Memory Devices - RAM, ROM, PROM, EPROM, EEPROM, PLD.

**Syllabus:**

**Boolean algebra and logic gates:**

Introduction to digital systems, basic Boolean theorems and properties, truth table, Boolean functions, canonical and standard forms, other logic operations **[2 Hours]**

**Simplification of Boolean functions** :

The map method: maps up to five variables, Product of sums and Sum of products simplification, NAND and NOR implementation. Don’t care conditions, the tabulation method, determination and selection of prime implicants. **[8 Hours]**

**Combinational logic :**

Design procedure, Design of Adders, Subtractors, code converters etc., Multilevel NAND and NOR circuits, EX-OR and equivalence functions. **[6 Hours]**

**Combinational logic with MSI and LSI:**

Application of typical TTL integrated circuit components like Binary Parallel adder[74283], carry look ahead adder [74182], multiplier using 74283, BCD adder [82583], Magnitude Comparator [7485], Decoders [74138,7442], Encoders [74148], Multiplexers [74157], De Multiplexers, ROMS and Programmable logic arrays [PLA-829 100] for designing combinational logic.

**[12 Hours]**

**Synchronous sequential logic:**

Different types of flip-flops and their triggering, Analysis of clocked sequential circuits, Design Of clocked sequential circuits-state reduction, state assignment, flip-flop excitation tables, Shift Registers, design of counters[Asynchronous and Synchronous],Shift register counters, Design of synchronous sequential circuits [Ex: sequence detectors] **[12 Hours]**

**Asynchronous sequential logic:**

Analysis Procedure, Circuits with latches and Design Procedures. **[3 Hours]**

**Memory Devices:**

RAM, ROM, PROM, EPROM, EEPROM, PLD. **[5 Hours]**

**Outcomes:**

After studying this course, students are able to:

* Identify the applications of various elements of digital system abstractions.
* Transform complex Boolean expressions using Boolean theorems, K-Map, tabulation methods.
* Design combinational and sequential logic circuits.
* Classify different memory devices of a computer.

**References:**

1. Mano M.R.,‎ Kime C.R.,‎ Martin T., *Logic & Computer Design Fundamentals (5e),* Prentice Hall India, 2015.
2. Tocci R.J., Widmer N.S., Greegory L.M., *Digital Systems: principles and Applications* *(12e),* Pearson Education India, 2017.
3. Wakerly J.F., *Digital Design Principles and Practices* *(4e),* Pearson Education, 2014.

**ICT 2155: OBJECT ORIENTED PROGRAMMING [3 1 0 4]**

**Objectives:**

* Understand the fundamentals of object oriented programming.
* Writing and enhancing classes, arrays, inheritance, polymorphism, abstract classes and interfaces, I/O streams, collections, exceptions and threads, swing basics.

**Abstract:**

Introduction to the java programming language, Importance of Java in the internet, Data types, Variable and arrays, Type conversion and casting, Operators and control statements, Classes and inheritance, Packages and interfaces, Collections Framework - array list, vector and dictionary, String handling, Exception handling, Thread concepts – synchronization, inter thread communication, Input/output – File:file input stream, File output stream, Random access files, Reader, Writer, Serialization, Serializable, Object input stream, Object output stream, Swings - swing fundamentals, Introduction to event handling.

**Syllabus:**

**Introduction**

The birth of modern programming language C, the need for C++, java, importance of java in the internet, java applets and applications, security, portability, the byte code. An overview of java, OOP, two paradigms, abstraction, the three OOP principles. **[2 Hours]**

**Data Types, Variable and Arrays**

Simple types, integers, floating point types, characters, booleans, variables – declaring variable, dynamic initialization, the scope and life time of variables, type conversion and casting, arrays-one dimensional arrays and multi-dimensional arrays. **[3 Hours]**

**Operators and control statements**

Arithmetic operators, bitwise operators, relational operators, logical operators, assignment operators, ternary operators, operator precedence. Control statements – if, switch, while, do-while, for nested loops, break, continue. **[4 Hours]**

**Classes and Inheritance**

Class fundamentals, declaring objects, assigning object references variables, introducing methods, constructors, overloading method, using objects as parameters, argument passing, returning objects, recursion, use of static and final key word, nested and inner class, using command line arguments. Inheritance – basics, using super, creating a multi-level hierarchy, when constructor are called, method overriding, dynamic method dispatch, using abstract classes, using final with inheritance, wrapper classes. **[8 Hours]**

**Packages and Interfaces**

Packages, defining a package, use of classpath, package example, access protection, importing packages, interfaces – defining an interface, implementing interfaces, applying interfaces, variables in interfaces, extending interfaces. **[5 Hours]**

**Array list and Vectors**

Collections overview, collection interface, the list interface, array list class, obtaining an array from an array list, vector, dictionary. **[4 hours]**

**String Handling**

String constructors, string length, special operations, character extraction, string comparison, searching strings, modifying a string, string buffer, string tokenizer. **[4 Hours]**

**Exception Handling**

Fundamentals, exception types, uncaught exception, using try and catch, multiple catch clauses, nested try statements, throw, throws, finally, java’s built in exception, creating exception subclasses, using exception. **[4 Hours]**

**Thread Concepts**

The java thread model, thread priorities, synchronization, thread class and runnable interface, the main thread, creating a thread, creating multiple threads, using is alive[] and join[], inter thread communication. **[5 Hours]**

**Input/Output**

Java I/O classes and interfaces, file – directories, using filename filter, the stream classes, the byte streams-input stream, output stream, file input stream, file output stream, byte array input stream, byte array output stream, random access files. The character streams- reader, writer, filereader, filewriter, bufferedreader, bufferedwriter. Serialization, serializable, externalizable, object output, object output stream, object input, object input stream. **[6 Hours]**

**Swings**

Swing fundamentals, writing swing application, swing library, layouts and controls. Introduction to event handling **[3 Hours]**

**Outcomes:**

By the end of this course, the student should be able to:

* Develop simple applications using Java primitives
* Implement OOP Concepts using JAVA
* Use inbuilt library packages of JAVA
* Develop Java application using object oriented concepts
* Write simple concurrent programs using threads

**References:**

1. Schildt H., *Java-The Complete Reference (9e)*, Tata McGraw-Hill 2014.
2. [Horstmann](https://www.amazon.in/Cay-S.-Horstmann/e/B000AQ1QDY/ref=sr_ntt_srch_lnk_1?qid=1519985819&sr=1-1) C. S. & Cornell G., *Core Java Volume I – Fundamentals (9e),* Prentice Hall 2013.
3. [Horstmann](https://www.amazon.in/Cay-S.-Horstmann/e/B000AQ1QDY/ref=sr_ntt_srch_lnk_1?qid=1519985819&sr=1-1) C. S. & Cornell G., *Core Java Volume II* *– Advanced Features* *(9e)*, Prentice Hall 2013.

**ICT 2156: PRINCIPLES OF DATA COMMUNICATION [3 1 0 4]**

**Objectives:**

* To understand basics of data communication
* To understand error detection and correction techniques
* To understand data link layer protocols
* To understand the performance of media access protocols

**Abstract:**

Introduction to Data Communication, Signals, Basic properties of data communication system, Nyquist rate, Shannon Capacity, Signal encoding and Tx and Rx models, Modulation schemes. Properties of Media and digital transmission systems, wired and wireless medium, Error detection and correction, Block codes, CRC, Hamming code, Stop and wait flow control, Sliding window flow control, ARQs, HDLC, Multiplexing, Media Access Sublayer and LAN, Approaches to sharing transmission medium, Random access protocols, Token passing protocols, IEEE LAN standards, Bridges, MAN, FDDI.

**Syllabus:**

**Data Communication fundamentals:** Introduction to Data Communication, Signals, Digital representation of information, Basic properties of data communication system, Time and frequency domain characterization of communication channels, Nyquist signaling rate, Shannon Channel capacity, Line coding-NRZ, bipolar, Manchester, Differential Manchester encoding, Modems and digital modulation- ASK, FSK, PSK, QAM. [**14 hours]**

**Properties of Media and Digital Transmission Systems:** Twisted pair, Coaxial cable, Optical fiber, Wireless transmission.  **[04 hours]**

**Error detection and correction:** Asynchronous and synchronous transmission, Error detection and correction basics, Parity check, Internet checksum, Polynomial codes, Block codes, Hamming code. [**08 hours]**

**Peer to Peer Protocols:** Peer to peer protocols and service models, ARQ protocols- Stop and wait, Go back N, selective repeat, Transmission efficiency of ARQ protocols, Other adaptation functions- Sliding window flow control, Timing recovery for synchronous services, Reliable stream service, Data link control- HDLC datalink control, point to point control. Multiplexing-FDM, TDM, STDM. [**14 hours]**

**Media Access sublayer and LAN**: Introduction to layered architecture, Protocols, Approaches to sharing transmission Medium, Random Access Protocols, Token Passing protocols, IEEE LAN standards, Bridges, MAN[IEEE802.6], FDDI. **[08 hours]**

**Course Outcome**

The students will be able to

* Outline the basics of data communication
* Compute frame check sequence and error correction codes
* Explain data link layer protocol
* Compute the performance of media access protocols

**References:**

1. Stallings W., *Data & Computer Communications* *(9e),* Pearson Education Inc., Noida, 2017.
2. Frozen B., *Introduction to data communication & networking* *(4e),* Tata McGraw Hill, New Delhi-2014.
3. Garcia A. L., Widjaja I., *Communication Networks (2e),* Tata McGraw Hill, 2011.

**ICT 2162: DATA STRUCTURES LAB [0 1 2 2]**

**Objectives:**

* To implement basic data structures.
* To apply the suitable data structure for the given real world problem
* To apply various sorting and searching techniques for the given problem

Application using arrays, String operations, Class concepts: Creation, Initialization using constructors, Applications of stacks, Arithmetic expression conversion and evaluation using stack, queues, Sparse matrix representation, Transpose of a sparse matrix, Singly linked lists and applications, Circular linked lists, Doubly linked lists, polynomial addition and multiplications using circular linked lists, Binary Tree: creation, deletion and traversal techniques, Binary search tree operations, sorting and searching techniques.

**Outcomes:**

By the end of this course, the students are able to:

* Identify suitable data structures for the given problem.
* Associate suitable searching and sorting techniques for the given data structure.
* Demonstrate the working of linear and non-linear data structure.

**References:**

1. Horowitz E, Sahni S., Mehta D., *Fundamentals of Data Structures in C++ (2e),* Golgotha Publications, 2008.
2. Weiss M. A., *Data Structures and Algorithm Analysis in C++ (3e),* Pearson Education, 2009.
3. Horowitz E., Sahni S., Anderson-Freed S.*, Fundamentals of Data structures in C (2e),* Silicon Press, 2008.

## **ICT 2163: DIGITAL SYSTEMS LAB [0 1 2 2]**

**Objective:**

* Introduce different tools for implementing digital circuits.
* To illustrate the implementation MSI combinational logic circuits using trainer kit.
* To demonstrate the working of sequential logic circuits using output waveforms.

**Abstract:**

Verification of Boolean algebra and De Morgan theorems, Simplification of Boolean expressions using K-maps, Combinational logic circuit implementation – Binary parallel adder [7483], BCD adder, Multiplier, Code converter, Comparator, 3 to 8 decoder [74138], Magnitude comparator [7485], Multiplexers [74151, 74153, 74157] ICs, Sequential logic circuits- Flip flops, Conversion of flip-flops, Analyzing timing diagram using output waveforms, Asynchronous and Synchronous counters [7490, 7493, 74193 ICs], Shift registers, Shift register counters, Sequence generators, Sequence detectors.

**Syllabus:**

* Verification of Boolean algebra and De Morgan theorems.
* Simplification of Boolean expressions using K-maps
* Combinational logic circuit design – Adders, Subtractors, multiplexers, Decoders, encoders, shift registers, code converters.
* Sequential logic circuit design - Asynchronous and Synchronous counters, Shift register counters, sequence detectors.

**Outcomes:**

After studying this course, students are able to:

* Simplify and verify the Boolean expression using basic/universal gates on trainer kit
* Design and build a combinational and sequential circuit using ICs
* Apply the digital system concepts to solve a given problem and build the circuit using MSI circuits on simulation tool

**References:**

1. Mano M.R.,‎ Kime C.R.,‎ Martin T., *Logic & Computer Design Fundamentals (5e),* Prentice Hall India, 2015.
2. Tocci R.J., Widmer N.S., Greegory L.M., *Digital Systems: principles and Applications* *(12e),* Pearson Education India, 2017.
3. Wakerly J.F., *Digital Design Principles and Practices* *(4e),* Pearson Education, 2014.

**ICT 2164: OBJECT ORIENTED PROGRAMMING LAB [0 0 3 1]**

**Objectives:**

* To implement the basic concepts of object oriented programming
* To provide working knowledge of abstraction, encapsulation, inheritance and polymorphism.

**Abstract:**

Programs based on the following concepts: Data types, Type conversions, Operators, Control statements, Classes, Inheritance, Polymorphism, Threads, Interfaces and abstract classes, Collections: arraylist and vector, String handling, File handling, Swings.

**Outcomes:**

* Implement Object Oriented Programming Concepts.
* Use and create packages and interfaces, collections, implement exception handling.
* Implement string programs, use input/output streams, create swings

**References:**

1. Schildt H., *Java-The Complete Reference (9e)*, Tata McGraw-Hill 2014.
2. [Horstmann](https://www.amazon.in/Cay-S.-Horstmann/e/B000AQ1QDY/ref=sr_ntt_srch_lnk_1?qid=1519985819&sr=1-1) C. S. & Cornell G., *Core Java Volume I – Fundamentals (9e),* Prentice Hall 2013.
3. [Horstmann](https://www.amazon.in/Cay-S.-Horstmann/e/B000AQ1QDY/ref=sr_ntt_srch_lnk_1?qid=1519985819&sr=1-1) C. S. & Cornell G., *Core Java Volume II* *– Advanced Features* *(9e)*, Prentice Hall 2013.

**FOURTH SEMESTER**

**MAT 2256: ENGINEERING MATHEMATICS-IV [2 1 0 3]**

Basic Set theory, Axioms of probability, Sample space, conditional probability, total probability theorem, Baye’s theorem One dimensional and two dimensional random variables, mean and variance, properties, Chebyschev’s in equality, correlation coefficient, Distributions, Binomial, Poisson, Normal and Chi square. Functions of random variables: One dimensional and Two dimensional, F & T distributions , Moment generating functions, Sampling theory, Central limit theorem, Point estimation, MLE, Interval estimation, Test of Hypothesis : significance level,certain best tests; Chi square test.

**References:**

1. P.L.Meyer, *Introduction to probability and Statistical Applications, (2e),* Oxford and IBH publishing, 1980
2. Miller, Freund and Johnson, *Probability and Statistics for Engineers*, (8e), PHI, 2011.
3. Hogg and Craig , *Introduction to mathematical statistics*, (6e), Pearson education, 2012
4. Ross Sheldon M, *Introduction to Probability and Statistics for Engineers and Scientists*, Elseveir, 2010

**ICT 2255: COMPUTER NETWORK PROTOCOLS [4 0 0 4]**

**Objectives:**

* To provide basic knowledge of networking technologies and network protocol concepts
* To understand the functions of each layer and gain knowledge in different applications that use computer networks.
* To provide the student with fundamental knowledge of the various aspects of computer networking and enable students to appreciate recent developments in the area
* To be familiar with contemporary issues in networking technologies.

**Abstract:**

Introduction to Computer Networks: Definition, Network Layer, Network Layer services, Interfacing - Bridges, IP addressing, Subnetting and Supernetting, IPv6 addressing, Delivery Forwarding, and Routing of IP Packets, Internet Protocol **-** Datagram, Fragmentation, Options, Checksum, Introduction to Routing Protocols, Interior and Exterior routing, Dynamic IP Routing Protocols - RIP, RIP Version 2, OSPF, Routing between peers – BGP, ARP and RARP, Internet Control Message Protocol, User Datagram Protocol, Transmission Control Protocol and Introduction to application layer, Domain Name System (DNS), DHCP, FTP, SNMP.

**Syllabus:**

**Introduction to Network Layer:**

Introduction, Switching, Circuit switching at network layer, Network Layer services, Interfacing - Bridges **[6 Hours]**

**IPv4 and IPv6:**

Classful addressing, Classless addressing, Subnetting, Masking, Variable length subnetting, supernetting, Special address, NAT, IPv6 addressing **[5 Hours]**

**Delivery and Forwarding of IP Packets**:

Direct and Indirect Delivery, Forwarding, Internet Protocol **-** Datagram, Fragmentation, Options, Checksum & IP Design. [**7 Hours]**

**Unicast Routing Protocols:**

Interior and Exterior routing, Dynamic IP Routing Protocols - RIP, RIP Version 2, OSPF, Routing between peers – BGP **[6 Hours]**

**ICMP, ARP and RARP:**

ARP and RARP, Internet Control Message Protocol **-** Types of messages, message format, error reporting, query, Checksum & ICMP Design. **[4 Hours]**

**User Datagram Protocol:**

Process-To-Process Communication, User datagram, UDP operation, Uses of UDP. **[3 Hours]**

**Transmission Control Protocol:**

TCP services, A TCP connection**,** State Transition Diagram, Flow control, Error Control, Congestion control, TCP Timer. **[8 Hours]**

**Application Layer Protocols:**

Introduction to application layer, Domain Name System (DNS) – Namespace, Resolution, DNS Messages, Types of Records, Host Configuration: DHCP , Remote login : Telnet, File Transfer – FTP, Network Management : SNMP.  **[9 Hours]**

**Course Outcomes:**

Upon completion of this course a student will be able to

* Illustrate the proper usage of various protocols that has been used in the different layers of TCP/IP protocol suite
* Interpolate the basic protocols of computer networks in network design and implementation.
* Apply various protocols to solve challenges in a given scenario.

**References:**

1. Forouzan B. A., *TCP/IP Protocol Suite (4e),* Tata McGraw Hill 2017.
2. Tanenbaum A. S., *Computer Network (5e),* Prentice Hall of India Pvt Ltd 2013.
3. Forouzan B. A., *Data Communications and Networking (5e),* Tata McGraw Hill 2013.
4. Garcia L., Widjala, *Communication Networks (2e),* Tata McGraw Hill 2004.

## **ICT 2256: COMPUTER ORGANIZATION AND MICROPROCESSOR SYSTEMS [3003]**

Microprocessor 8086 Architecture, Pin diagram, Modes of operation, Segmentation and memory addressing, Addressing modes, Assembler directives, Assembly language development tools, Instruction set, Stacks and subroutine, Macros and procedures, Assembly language programming, Interrupts, BIOS and DOS interrupts, Basic IO interfacing- 8255 Programmable Peripheral Interface, 8254 Programmable Interval Timer, 8259 Programmable Interrupt Counter, Computer Organization: Introduction, Execution Unit - Combinational shifter design, Adders, Arithmetic and Logic Unit design, Multiplication algorithms, Division algorithms., Control Unit- Introduction, Basic concepts, Hardwired and Micro programming approach, Memory Unit, Input & Output.

**Objectives:**

* To understand the architecture of 8086 microprocessor
* Study the instruction set, various interrupts, macros, procedures
* To write the programs using instructions
* Learn to design an ALU, Execution unit, control unit
* Understand memory unit and input/output

**Syllabus:**

#### 8086 Architecture:

Intel 8086 based micro computer system, 8086 architecture, and programmers’ model, and functional pin diagram, modes of operation, segmentation and memory addressing. **[3 Hours]**

**Instruction Set:**

Addressing modes, assembler directives.

Instruction types: Data movement instruction, arithmetic and logic instruction, process control instructions, string instructions, branch instructions and related programs. **[9 Hours]**

**Development Tools:**

Assembly language development tools, stacks and subroutine, macros and procedures and related programs **[3 Hours]**

**Interrupts:** BIOS and DOS interrupts and related programs. **[3 Hours]**

**Basic I/O interfacing:** 8254, 8255, 8259- Architecture and interfacing **[3 Hours]**

**Computer Organization:**

**Introduction:**

Evolution of computers, Von- Neumann architecture, Computer structures: General register machine, Accumulator based machine, stack machines. Introduction to RISC and CISC architecture.

**[2 Hours]**

**Execution Unit:**

Combinational shifter design, Adders. Arithmetic and Logic Unit Design, multiplication algorithms, division algorithms. **[3 Hours]**

**Control Unit:**

Introduction, basic concepts, Design methods: Hardwired and Micro – programming approach.

**[3 Hours]**

**Memory Unit:**

Types of memory and characteristics, memory hierarchy, main memory design, the cache memory and mapping techniques **[4 Hours]**

**Input & Output:**

Programmed I/O, Interrupt I/O, direct memory access, I/O bus standards. **[3 Hours]**

**Outcomes:**

* Recall 8086 architecture
* Write assembly language programs using development tools
* Understand the interfacing of programmable devices to 8086 microprocessor
* Understand the organisation of various parts in computer system
* Design building blocks of computer system

**References:**

1. Hall D.V., *Microprocessors and Interfacing: Programming and Hardware (3e),* Tata McGraw Hill, 2017, **ISBN-10:** 9781259006159
2. Brey B.B., *The Intel Microprocessors: 8086 to Pentium Pro - Architecture, Programming and Interfacing (8e),* Prentice Hall of India, 2012
3. Udaykumar K, Umashankar B.S., *Advanced microprocessors and IBM –PC assembly language programming*, McGraw Hill Education, 2017.
4. Rafiquzzaman M and Rajan C., *Modern computer Architecture,* Galgotia Publications Pvt. Ltd, 2012.

**ICT 2257: DESIGN AND ANALYSIS OF ALGORITHMS [3 1 0 4]**

**Objectives:**

* To analyse asymptotic performance of algorithms.
* To study important algorithmic design paradigms.
* To familiarize with specific algorithms for a number of important computational problems.
* To understand the different P, NP, NP complete and NP hard problems.

**Abstract:**

Introduction, Graphs: Representation of graphs & Digraphs, Graph Search Methods: Breadth First Search, Depth First Search, Shortest path algorithms, Algorithm Design Techniques: The Greedy Method, Divide and Conquer, Dynamic Programming, Tree-Binary search trees, Heap Trees, Height Balanced Tree, B Trees, B+ trees. Hashing: hash functions, collision resolution techniques. Heaps and priority Queues, P, NP, NP-Complete and NP hard problems, Approximation Algorithms.

**Syllabus:**

**Introduction:**

Space and Time complexity, Asymptotic notations **[4 Hours]**

**Graphs:**

Definitions, Applications, Properties, The Graph & Digraph as ADTs, Representation of graphs and Digraphs, Breadth First Search, Depth First Search, Finding a path, Connected Graphs & Components, Spanning trees. **[6 Hours]**

**Greedy method:**

Optimization problems, container loading, Fractional Knapsack problem, Topological sorting, Bipartite cover, Single-Source Shortest paths, Minimum cost spanning Trees, Kruskal’s Algorithm, Prim’s Algorithm **[7 Hours]**

**Divide and Conquer:**

Minimum and Maximum, Strassen’s matrix multiplication, Merge Sort, Quick sort, Selection problem, Closest pair of points, Solving Recurrence Equations **[6 Hours]**

**Dynamic Programming:**

0/1 Knapsack problems, Matrix Multiplication Chains, All pair’s shortest paths.  **[5 Hours]**

**Backtracking & Branch and Bound:**

0/1 Knapsack problem, Max clique and Travelling salesperson. **[6 Hours]**

**Trees:**

Binary search trees, Heap Trees, Height Balanced Tree, B Trees, B+ trees, Red Black Trees, Splay Tree, Tries **[6 Hours]**

**Hashing Techniques:**

Hash function, Address calculation techniques, Common hashing functions, Collision resolution techniques, open addressing, closed addressing, separate chaining, Linear probing, Quadratic probing, double hashing. **[4 Hours]**

**NP-Completeness and Approximation Algorithms**

Polynomial Time and verification, P and NP Problems, NP-Completeness and Reducibility, NP-Hard problems, Approximation Algorithms for Vertex-cover problem and traveling salesperson problem. **[4 Hours]**

**Outcomes:**

By the end of this Lab, the student are able to:

* Understand asymptotic notations to represent the complexities of algorithms.
* Understand the basic concepts of graph traversal methods.
* Apply various algorithm designing techniques for a given problem.
* Comprehend the basic concepts of trees and hashing techniques.
* Understand NP complete and NP hard problems.

**References:**

1. Cormen T H., Leiserson C. E., Rivest R. L., Stein C., *Introduction to Algorithms* *(3e),* Prentice-Hall India, 2009.
2. Sahni S., *Data Structures, Algorithms and Applications in C++* *(2e),* Silicon Press, 2005.
3. Weiss M., *Data Structures and Algorithm Analysis in C* *(3e),* Pearson Education, 2009.

## **ICT 2258: OPERATING SYSTEMS [3 1 0 4]**

**Objectives:**

* To familiarize with the basic functionality and the evolution of different types of operating systems.
* To Learn and understand various algorithms related to CPU scheduling, deadlocks, memory management, and storage management.
* To learn basic aspects of real time operating systems

**Abstract:**

Introduction to Operating systems - Operating System Services, Operating system Structure, System calls, Process management - Process concept, Threads, Inter-process communication, CPU Scheduling, Process synchronization, Handling deadlocks – Deadlock Characterization, Deadlock Detection, Prevention, Avoidance and Recovery, Memory management - Main memory, Swapping, Contiguous Memory Allocation, Paging, Segmentation, Virtual memory – Demand Paging, Page Replacement, Thrashing, Allocating Kernel Memory, Storage Management- File management, Disk scheduling, Case study on Unix based Operating system – Design Principles, Kernel Modules, Basic concepts of Real time operating systems – Classification of Real Time Systems, Microkernels, Scheduling.

**Syllabus:**

**Introduction:**

Operating system structure, Operating system operations, Distributed systems, Special purpose systems, Computing environments, Open source operating systems. **[3 hours]**

**CPU Scheduling:**

Process concepts: Process states, Process control block, Scheduling queues, Schedulers, Context switch, Multi-threaded programming: Overview, Multithreading models, Threading issues, Process scheduling: Basic concepts, Scheduling criteria, scheduling algorithms. [**7 hours]**

**Process Synchronization:**

Synchronization: The Critical section problem, Synchronization hardware, Semaphores, Classic problems of synchronization, monitors.  **[6 hours]**

**Deadlocks**:

Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock. **[8 hours]**

**Memory management:**

Memory management strategies, Swapping, Contiguous memory allocation, Paging, Structure of the page table, Segmentation. **[6 hours]**

**Virtual Memory:**

Demand paging, copy on write, page replacement, allocation of frames, thrashing.

**[7 hours]**

**Storage Management:**

File concept, Access methods, directory structure, file system structure, directory implementation, allocation methods, free space management, disk structure, and disk-scheduling **[5 hours]**

**Case study on UNIX based Operating system:**

Design principles, Kernel modules, Process management, Memory management. **[2 hours]**

**Real time systems:** Characteristics of Real time operating systems, classification of real time systems, Micro kernels and RTOS, scheduling in RTOS, Rate monotonic scheduling, EDF, Priority inversion **[4 hours]**

**Outcomes:**

At the end of the course, the students are able to

* Acquire detailed understanding of operating system functionalities.
* Apply the knowledge to solve issues in process as well as memory management.
* Able to understand the fundamental concepts of real time operating systems.
* Apply the knowledge to understand modern operating systems concepts.

References:

1. Silberschatz A., Galvin P.B. & Gagne G*., Operating System Concepts (9e),* Wiley, 2012*.*
2. Stallings W., *Operating Systems: Internals and Design Principles (9e),* Pearson, 2017*.*
3. Laplante P.A. & Ovaska S.J., *Real time systems design and analysis (4e),* Wiley, 2013*.*
4. Mall R., *Real time systems: Theory and Practice (2e),* Pearson, 2009.

**\*\*\* \*\*\*\*: OPEN ELECTIVE-I**

**ICT 2263: ALGORITHM LAB [0 0 3 1]**

**Objectives:**

* To implement basic algorithm designing techniques.
* To understand basic approximation algorithm.
* To apply the appropriate designing technique for a given problem.

**Abstract:**

Graphs: Finding a Path, cycle. Greedy Method: 0/1 Knapsack Problem, Dijkstra’s Algorithm, Minimum spanning tree using Prim’s/Kruskal’s Algorithm. Divide and Conquer Method: Merge Sort, Quick Sort, Strassen’s Matrix multiplication method, Binary Search, Closest Pair of points. Dynamic Programming: Matrix Multiplication Chain Problem, 0/1 Knapsack problem, All pairs shortest path. Backtracking/Branch and Bound: Travelling sales person problem, 0/1 Knapsack Problem. Approximation Algorithm: Travelling sales person problem, Vertex cover problem.

**Outcomes:**

By the end of this Lab, the student are able to:

* Implement an algorithm to find path between any two vertices in the given graph.
* Apply the knowledge of shortest path algorithms for real world problems.
* Implement Greedy, Divide and Conquer, Dynamic Programming, Back tracking and Branch and Bound techniques to solve different problems.
* Implement approximation algorithm for travelling sales person and vertex cover problem.

**References:**

1. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C, *Introduction to Algorithms* *(3e),* Prentice- Hall India, 2009.
2. Sahni S., *Data Structures, Algorithms and Applications in C++ (2e),* Silicon Press, 2005.

**ICT 2264: MICROPROCESSOR SYSTEMS LAB [0 0 3 1]**

**Objectives:**

* To gain knowledge of assembly language programming
* To familiarise BIOS and DOS interrupts
* To understand the importance of Video RAM in advanced screen processing

**Abstract:**

Assembly language programs related to memory access – Block transfer, Addition, Subtraction, Multiplication, Division, Unpacked BCD arithmetic, Packed BCD arithmetic, Sorting, Searching, Code conversion, GCD, LCM, Recursive functions, Programs using DOS and BIOS interrupts to input from keyboard and display, Programs using string instructions, Menu driven programs, Programs using Video RAM.

**Outcomes:**

**After completing the course students will be able to**

* Use assembly language development tools
* Extend the knowledge of instruction set for writing efficient programs
* Use subroutine concepts in programming
* Write the programs using BIOS and DOS interrupts
* Apply the knowledge of advanced screen processing concepts

**References:**

1. Hall D.V., *Microprocessors and Interfacing: Programming and Hardware (2e),* Tata McGraw Hill, 2006.
2. Brey B.B., *The Intel Microprocessors: 8086 to Pentium Pro - Architecture, Programming and Interfacing (8e),* Prentice Hall of India, 2012.
3. Udaykumar K, Umashankar B.S., *Advanced microprocessors and IBM –PC assembly language programming*, McGraw Hill Education, 2017.
4. Rafiquzzaman M and Rajan C., *Modern computer Architecture,* Galgotia Publications Pvt. Ltd, 2012.

**ICT 2265: OPERATING SYSTEMS LAB [0 0 3 1]**

**Objectives:**

* To execute shell scripts in UNIX based operating system.
* To implement inter process communication using system calls.
* To implement algorithms for CPU scheduling as well as process synchronization

**Abstract:**

UNIX based operating system commands, executing shell scripts, inter process communication using system calls, implementing CPU scheduling algorithms, memory and deadlock management.

**Outcomes:**

**At the end of the course students are able to**

* Understand the working of UNIX based operating system.
* Illustrate the process management in operating systems.
* Implement CPU scheduling as well as synchronization algorithms
* Implement algorithms used to understand the functionality of modern operating systems.

**References:**

1. Blurn R.& Bresnahan C., *Linux Command Line Shell Scripting Bible, Wiley* *(3e),* 2015.
2. Silberschatz A., Galvin P.B.& Gagne G*., Operating System Concepts (9e),* Wiley, 2012*.*

**FIFTH SEMESTER**

**HUM 3052 : ESSENTIALS OF MANAGEMENT [2103]**

**Course Outcomes:**

On completion of this course the student should be able to:

* Understand the roles of managers, principles of management, managerial skills, and strategies required to run a business successfully.
* Develop an organizational structure and plan for manpower in a given business organization.
* Apply leadership and motivational theories in the organizational contexts
* Acquire budgetary skills through process and techniques of controlling
* Understand the various global managerial practices, professional ethics, corporate social responsibilities and entrepreneurship.

**Abstract: Yet to be received**

**Introduction to Management**

Introduction to Business, Industrial Business, Classification of Industries and Job Opportunities (referring the industries visiting our campus). Who is a manager? brief discussion on classification of managers (i.e., top, middle, and lower), managers and administrators, definitions of management. Functions of Managers/Management and time spent on various managerial functions by managers at various levels, two characteristics of managerial functions, Efficiency and Effectiveness. 14 Principles of Management by Henri Fayol. Three types of managerial responsibilities. Managerial Skills, roles played by managers (as per Mintzberg), Systems Approach to Management. **[6]**

**Planning**

Types of Planning: General and Managerial; Strategic, Tactical and Operational. Nature and characteristics of planning. Types of Plans: Mission, Vision, Goals, Objectives, Policies, Strategies, Rules, Procedures and Programs. Steps in Planning. Management By Objectives: Process, Key Result Areas, guidelines in setting the objectives, qualitative and quantitative objectives, Stakeholders, and their interests, Fiscal and Social Responsibilities. Strategic Planning: Planning Tools – SWOT, TOWS, Business Portfolio Analysis and Porter’s model; Process. **[5]**

**Organizing**

Process, Principles of Organizing; Span of Control, factors affecting the span of control and levels. Departmentation: meaning, Types of Departmentation – by Function, Product, Customer, Geography, Time, Process, Simple numbers, and combination. Types of managers – line, staff, functional and general. Structures – Line, Staff, Line, and staff, Functional, Committee, Project and Matrix. Delegation – process and principles; Delegation and decentralization. Formal and Informal organization. **[4]**

**Staffing**

HRM and HRD, in brief – Job Analysis, Job Description and Job Specification; Recruitment – methods and sources; selection – process, techniques and instruments; Induction and Orientation. Systems Approach to Staffing; Managerial development and training programmes. **6]**

**Leading**

Meaning, differences between – leading and managing, leader and manager; motives, motivation, and motivators. Theories of motivation – Maslow’s Need Hierarchy, Herzberg’s 2 – factor theory and McGregor X and Y theory; Motivational techniques. Leadership – theories, ingredients, and styles based on authority, Likert’s 4 systems of management and Leadership Grid. Communication – general and managerial, types of communication, barriers of communication. **[7]**

**Controlling**

Process, Types of Control, Management Control Techniques – budgetary, non-budgetary and network; Types of budgets – financial, operational, and non-monetary. Critical point Control Technique and standards. Direct and Preventive Controls. **[2]**

**Entrepreneurship, International Management Practices, Professional Ethics and Global Issues**

Functions of an entrepreneur, types of entrepreneurs; traits and characteristics of entrepreneurs and entrepreneurs. Business Plan preparation. Nature, and purposes of international businesses and Multinational Corporations; Managerial practices in Japan and USA, and Theory Z. Professional Ethics - Senses of Engineering Ethics, Variety of moral issues, Types of inquiry, Moral dilemmas, Moral Autonomy, Kohlberg’s theory, Gilligan’s theory, Consensus and Controversy, Models of professional roles, Theories about right action, Self-interest, Customs and Religion, Uses of Ethical Theories. Global Issues - Multinational Corporations, Environmental Ethics, Computer Ethics, Weapons Development, Engineers as Managers, Consulting Engineers, Engineers as Expert Witnesses and Advisers, Moral Leadership, Code of Conduct, Corporate Social Responsibility. **[6]**

**References**

1. Harold Koontz & Heinz Weihrich (2012), “Essentials of Management”, McGraw Hill, New Delhi.
2. Peter Drucker (2004), “The practice of management”, Harper and Row, New York.
3. Vasant Desai (2007), “Dynamics of entrepreneurial development & management”, Himalaya Publishing House.
4. Poornima M Charantimath (2006), “Entrepreneurship Development”, Pearson Education.
5. Mike W. Martin & Ronald Schinzinger (2003), “Ethics in engineering”, Tata McGraw Hill, New Delhi.
6. Govindarajan M, Natarajan S, & Senthil Kumar V S (2004), “Engineering Ethics”, Prentice Hall of India, New Delhi.
7. R. S. Nagarazan. (2004), “A text book on professional ethics and human values”, New age international publishers, New Delhi.

**ICT 3156: CYBER SECURITY [3 0 0 3]**

**Objectives:**

* To learn the basics of cyber security.
* To identify attacks on operating system, network and web.
* To be aware of the state-of-the-art technologies used for cyber security through real-world case studies.
* To analyze the security vulnerabilities in the existing systems.

**Abstract:**

Introduction to Computer Security, Toolbox: Authentication, Access Control and Cryptography, Programming Insights- Non-malicious programs, Malicious Programs, Viruses, Worms, Trojans, Countermeasures, Hacking- Basics of hacking, Phishing, Brute Force Attack, Denial of Service, Distributed Denial of Service Attacks, Penetration Testing, Bots and Botnets, Attacks on The Web, Operating Systems and Networks, Security Countermeasures- Browser Encryption, Onion Routing, IP Security Protocol Suite (IPsec), Virtual Private Networks, Firewalls, Intrusion Detection and Prevention Systems, Network Management, Management, Incidents, Ethics, Case Studies on Cyber Crime and Cyber Terrorism.

**Syllabus:**

**Introduction to Computer Security:**

Basics of Computer security, Confidentiality, Integrity, Availability, Threats, Harms, Vulnerabilities, Controls, Conclusion **[2 hours]**

**Toolbox- Authentication, Access Control and Cryptography:**

Authentication- Identification Vs Authentication, Authentication Based on Phrases and Facts,

Authentication Based on Biometrics, Authentication Based on Tokens, Federated Identity Management, Multifactor Authentication, Secure Authentication, Access Control- Access Policies, Implementing Access Control, Existing Access Control Models

Cryptography- Terminology, Symmetric and Asymmetric Encryption- AES, DES, RSA, Message Digests, Key Exchange, Certificates, Digital Signatures **[6 hours]**

**Programming Insights- Non-malicious vs malicious programs:**

Non-malicious programs, Malicious Programs- Viruses, Worms, Trojans, Countermeasures

**[4 hours]**

**Hacking:**

Basics of hacking, Phishing, Brute Force Attack, Denial of Service, Distributed Denial of Service Attacks, Penetration Testing, Bots and Botnets [**4 hours]**

**Attacks on the Web, Operating Systems and Networks:**

Web: Browser Attacks, Web Attacks targeting Users, Obtaining Users or Website Data, Email Attacks

Operating System: Security in Operating System, Security in the Design of Operating Systems, Rootkit.

Networks: Network Security Attacks, DoS, DDoS **[6 Hours]**

**Security Countermeasures:**

Browser Encryption, Onion Routing, IP Security Protocol Suite (IPsec), Virtual Private Networks, Firewalls, Intrusion Detection and Prevention Systems, Network Management

**[4 hours]**

**Management, Incidents, Ethics:**

Security Planning, A Measurement Primer for Cybersecurity, Handling Incidents, Risk Analysis, Risk Matrices, Lie Factors, Misconceptions, and Other Obstacles to Measuring Risk, Cyber Crime & Cyber Terrorism: Definitions, Emerging Threats, Ethical Issues in Computer Security, Incident Analysis with Ethics, **[6 hours]**

**Case Studies:**

Case Studies on Cyber Crime & Cyber Terrorism **[4 Hours]**

**Outcome:**

After completion of the course students are able to

* Understand the basics of cyber security.
* Evaluate attacks on operating system, network and web.
* Analyze the existing vulnerabilities and propose solutions.
* Examine real case studies of cyber security incidents and their mitigation.

**References:**

1. Pfleeger C. P., Pfleeger S. L. and Margulies J., *Security in Computing (5e),* Prentice Hall, 2015.
2. Akhgar B., Staniforth A. and Bosco F., *Cyber Crime and Cyber Terrorism Investigator's Handbook (1e),* Syngress Publishing, 2014.
3. Hubbard D. W. and Seiersen R*., How to Measure Anything in Cybersecurity Risk,* John Wiley & Sons, 2016.
4. Mitnick K. D. and Simon W. L., *Art of Intrusion,* Wiley Publishing Inc. 2005.
5. Singer P. W. and Friedman A*., Cybersecurity and Cyber war- What Everyone Needs to Know,* Oxford*.*

**ICT 3157: DATABASE SYSTEMS [3 0 0 3]**

**Objectives:**

* To understand the concepts of database system.
* To manage the relational database using SQL and PL/SQL constructs.
* To design conceptual database.
* To gain understanding of database transactions and concurrency control.
* To understand unstructured databases.

**Abstract:**

Introduction to database system, Database users, Database architecture, Relational database, Keys, Schema, Formal relational query language, SQL basics, Constraints, Intermediate SQL, Joins, Nested queries, Advanced SQL, Functions, Procedures, Triggers, High level data modelling using entity relationship model, Relational database design, Notion of functional dependencies, Normalization, Transaction management, ACID properties, Serializability, Concurrency control, Locking, Deadlock handling, Unstructured database, Introduction to NoSQL, Basics of document-oriented database, MongoDB.

**Syllabus:**

**Database and database users:**

Introduction to the relational database, view of data, advantages of using a DBMS, actors on the scene, database architecture.  **[2 Hours]**

**Relational Databases:**

Introduction to the Relational Model**,** Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, Relational algebra. **[3 Hours]**

**Introduction to SQL:**

Overview of the 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. **[5 Hours]**

**Intermediate SQL:**

Join expressions, Views, materialized views, Transactions, Integrity Constraints, SQL Data types and schemas, Authorization. **[2 Hours]**

**Advanced SQL:**

Introduction to PL/SQL, cursors, Functions and procedures, Triggers, Recursive queries. **[3 Hours]**

**Database design and ER model:**

Overview of the design process, the entity relationship model, constraints, removing redundant attributes in entity sets, entity relationship diagrams, ER design issues, Extended ER features.

**[3 Hours]**

**Relational database design:**

Features of good relational designs, Atomic domains and first normal form(1NF), second normal form(2NF), Decomposition using functional dependencies, 3NF, BCNF, Functional dependency theory: closure of functional dependency set/attribute sets, Armstrong’s axioms, canonical cover, extraneous attributes, dependency preservation, Algorithms for decomposition. **[8 Hours]**

**Transaction management**

Transaction concept, A 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 statement. **[4 Hours]**

**Concurrency control:**

Lock based protocols, deadlock handling, Multiple granularity, Timestamp based protocols, Validation based protocols. **[3 Hours]**

**Unstructured database:**

Introduction to NoSQL, RDBMS vs NoSQL, CAP theorem, Types of NoSQL databases, basics of MongoDB. **[3 hours]**

**Outcomes:**

The students are able to:

* Understand the database concepts.
* Apply procedural and non-procedural language constructs to manage database system.
* Design database using data modelling tool and normalization concepts.
* Describe transaction management and concurrency control concepts.
* Interpret the unstructured databases.

**References:**

1. Silberschatz A., Korth H. F., Sudarshan S., *Database system concepts* *(6e)*, McGraw-Hill, 2013.
2. Elmasri, Ramez, Navathe S., *Fundamentals of database systems* *(7e)*, Pearson, 2016.
3. Molina, Hector, Ullman J. D., Widom J., *Database systems, The Complete Book* *(2e),* Pearson Prentice Hall, 2013.
4. Chodorow K., MongoDB: *The definitive guide* *(2e)*, O’Reilly, 2013.

## **ICT 3158: EMBEDDED SYSTEMS [3 1 0 4]**

**Objectives:**

* To develop an understanding of the technologies behind the embedded computing systems
* To understand real world interfacing to micro controller
* To design software for embedded applications.

**Abstract:**

An overview of ARM-Cortex- M Architecture, CISC versus RISC, The RISC and ARM design philosophy, ARM addressing modes, Data transfer instructions, Arithmetic and logical instructions, Shift and rotate instructions, Branch and conditional branch instructions, Function call and return, Stack, Recursive functions, Conditional execution, Assembly language programming, Input/output I/O) programming, Timer/counter programming, I/O interfacing : LED, LCD, Keyboard, Stepper motor, ADC, and DAC, PWM, UART, Hardware and software synchronization, Multithreading, Nested Vectored Interrupt Controller (NVIC), External hardware interrupts, IO interrupts, SysTick interrupts.

**Syllabus:**

**Introduction to Embedded Systems and ARM Cortex-M Microcontroller:**

Embedded System, Computer Architecture, microprocessor versus microcontroller, CISC vs. RISC, choosing a microcontroller, ARM Cortex M Architecture, General purpose and Special Function Registers (SFRs), CPSR, System control block, ARM memory map, memory management, debugging **[6 Hours]**

**Assembly language programming:**

Addressing modes, data transfer instructions, arithmetic and logical instructions, shift and rotate instructions, branch and conditional branch instructions, function call and return, stack, recursive functions, conditional execution, assembly language programs **[8 Hours]**

**Input/output (IO) programming:**

Pin connect block, Pin function select registers, General Purpose Input and Output (GPIO) registers, GPIO configuration, GPIO programming using ARM C language, Interfacing: LEDs, Seven segment, multiplexed seven segments, LCD, keyboard, DC motor, Stepper motor

**[10 Hours]**

**Timer/ Counter programming:**

Timer versus counter, timer registers, timer architecture and operation, PWM timer and architecture, timer/counter programming, PWM programming. **[8 Hours]**

**Serial, ADC and DAC Interfacing:**

General introduction to serial interfacing, RS232, MAX 232, UART, UART programming, data acquisition system, Analog to Digital Converter (ADC), ADC registers, Digital to Analog converter (DAC), DAC registers, ADC and DAC programming. **[8 Hours]**

**Interrupt programming:**

Hardware and software synchronization, multithreading, Nested Vectored Interrupt Controller (NVIC), external hardware interrupts, IO interrupts, SysTick interrupts, timer/counter interrupts, ADC and DAC interrupts, UART interrupts, interrupt programming. **[8 Hours]**

**Outcome:**

Upon successful completion of this course student is able to

* Familiarise with the salient features of embedded systems.
* Illustrate the architecture of ARM Cortex- M microcontroller.
* Familiarise with the efficient software design for embedded systems.
* Outline software development for ARM Cortex-M microcontroller.
* Design real world systems using ARM Cortex-M microcontroller

**References:**

1. Jonathan W.V., *Embedded systems: Real-time interfacing to ARM Cortex-M microcontrollers* *(4e),* Createspace Independent Publishing Platform, June 2014.
2. Wilmshurst T., *Fast and Effective Embedded System Design applying the ARM mbed*, Elsevier, 2017.
3. Jonathan W.V., *Embedded systems: Introduction to Arm(r) Cortex-M Microcontrollers(5e),* CreatespaceIndependent publishing platform*,* June 2017.
4. UM10360, LPC 176x/5x *User Manual, NXP Semiconductors*, Rev. 3.1, 2014.
5. Joseph V., *A definitive Guide to ARM Cortex-M3 and Cortex-M4 processors (3e),* Elsevier, 2014.

**ICT 3159: SOFTWARE ENGINEERING [3 1 0 4]**

**Objectives:**

* To learn the Concepts of Software Engineering
* To understand the Software Development Life Cycle
* To learn the basic principles behind software configuration and risk management
* Model software requirements for application development
* To learn architectural, design patterns for design of complex, scalable software systems

**Abstract:**

Introduction to Software Engineering, Process Models- Agile Process Model, Requirement Engineering, Requirement Modeling- Scenario and Class based modeling, Design models and methodologies, Case study, Architectural and Design Patterns, Software Testing Strategies and Testing Techniques, Software Configuration Management and Risk Management, Introduction to Project Management- basics of product metrics, planning and scheduling

**Syllabus:**

**Introduction to software engineering:**

Software and software engineering, The changing nature of the software, Legacy software, Software Myths [**1 Hour]**

**Software Engineering Process Models:**

Phases in software development, The Waterfall Model, Evolutionary Models, Incremental Models, Specialized Models, Unified Models, Agile Process Model: Scrum.  **[6 hours]**

**Modeling:**

Understanding Requirements, Requirements Modeling: Scenario based Modeling and Class Based Modeling **[7 Hours]**

**Design:**

Design Process and Concepts, Design Model- Data design elements, Architectural design elements, Interface elements, Component level elements, Deployment level design elements, Pattern based design, Design Methodologies- Object Oriented Methods, Functional Methods. Case Study: Design/ model a software application **[10 Hours]**

**Architectural and Design Patterns:**

Introduction, Patterns, Design Patterns: Structural Decomposition, Organization of Work, Access Control, Management Communication, Architectural Patterns: Basic, Distributed Systems, Interactive Systems, Adaptable Systems **[8 hours]**

**Software Testing Strategies and Testing Techniques**- Strategic approach, Issues, Test strategies for Object oriented system, Levels of testing- Unit testing, Integration Testing, Validation testing, System testing, Debugging, Black Box testing, White box testing techniques. Case study: Testing the different use cases using different testing methods and testing tools. **[8 Hours]**

**Software Configuration Management and Risk Management-** The SCM Scenario, SCM Repository, SCM Process, Configuration management for web and mobile apps, Reactive vs. Proactive Risk, Risk Identification, Projection, Refinement and Management. **[4 Hours]**

**Introduction to Project Management-** Management Spectrum, Product, Process, People, Project Metrics, Project Planning Process, Basic principles of project scheduling. **[4 Hours]**

**Outcomes:**

**After completing the course student will be able to,**

* Understand the basics of software development life cycle
* Understand the basic principles behind software configuration and risk management
* Explore the importance of requirement analysis through scenario based exercise
* Adapt software design strategies using object oriented concepts
* Identify a suitable testing strategy to validate a given software application

**References:**

1. Pressman R. S., *Software Engineering A practitioner’s approach* *(8e)*, McGraw Hill, 2014.
2. Booch G., Rumbaugh J., Jacobson I., *The Unified Modeling Language User Guide* *(2e)*, Pearson, 2015.
3. Somerville I., *Software engineering* *(10e)*, Pearson Education, 2017.

**\*\*\* \*\*\*\*: OPEN ELECTIVE-II**

## **ICT 3163: DATABASE SYSTEMS LAB [0 1 2 2]**

**Objectives:**

* To get acquainted with front end design and connecting front end to the database using Visual C#.
* To understand the use of SQL as a data definition language, data manipulation language, and Transaction control language.
* To provide a strong formal foundation on implementation of views, triggers, cursors and procedural interfaces to SQL (Procedures and Functions).
* To comprehend the need and importance of NoSQL for unstructured data.
* To work productively in a team to design and implement a relational database project that provides a computing solution to a real world problem.

**Abstract:**

Basics of Visual C# for GUI design and control, Data Definition Language, Basic database query operations, Nested subqueries, Join Operations ,Views, Stored procedures, Functions, Trigger, Cursors, Transaction control queries, Data Access from Visual C#, Introducing NoSQL-MongoDB ,Design and development of application based on database concepts.

**Outcomes:**

The students are able to:

* Implement a graphical user interface (GUI) and access database through GUI using an integrated development environment.
* Design, create and query the relational database using structured query language.
* Demonstrate the working of procedural and non-procedural language.
* Perform create, read, update and delete operations on MongoDB
* Implement a mini project that provides solution to a real world problem.

**References:**

1. Ivan Bayross ,SQL, PL/SQL: The Programming Language of Oracle ,4th Revised Edition, BPB Publications,2010
2. https://www.homeandlearn.co.uk/csharp/csharp.html
3. https://www.tutorialspoint.com/csharp
4. https://www.javatpoint.com/oracle-tutorial
5. https://www.tutorialspoint.com/oracle\_sql/index.asp
6. <https://www.w3schools.com/sql/>
7. <https://docs.mongodb.com/manual/tutorial/>
8. <https://www.tutorialspoint.com/mongodb/mongodb_environment.htm>

**ICT 3164: EMBEDDED SYSTEMS LAB [0 0 3 1]**

**Objectives:**

* To develop skills in real world interfacing circuits
* To efficiently design software for embedded systems

**Abstract:**

Familiarization of data transfer from code segment to data segment and from data segment to data segment, Arithmetic operations, Logical instructions, Branch instructions, Code conversion from hexadecimal to decimal and decimal to hexadecimal, Packing and unpacking of ASCII digits, Sorting using selection sort and bubble sort techniques, Searching using linear and binary search techniques, Recursion, I/O interfacing of LEDs, LCD, keyboard, 7 segment display, Stepper motor, DAC and ADC, PWM. In addition to the above list of experiments, students are required to develop a mini project.

**Outcomes:**

* Recall the basics of an embedded system
* Program an embedded system using ARM controller
* Design, implement and test an embedded system.

**References:**

1. Jonathan W. Valvano “Embedded systems: real-time interfacing to ARM Cortex-M microcontrollers” Createspace Independent Publishing Platform, 4th edition, June 2014.
2. Toulson and Tim Wilmshurst. “Fast and Effective Embedded System Design applying the ARM mbed”, Elsevier, 2017.
3. Jonathan W. Valvano “Embedded systems: Introduction to Arm(r) Cortex-M Microcontrollers”, Createspace Independent publishing platform, 5th edition, June 2017.
4. UM10360, LPC 176x/5x User Manual, NXP Semiconductors, Rev. 3.1, 2014.
5. Joseph Viu, “A definitive Guide to ARM Cortex-M3 and Cortex-M4 processors, 3rd Edition, Elsevier, 2014.

**ICT 3165: NETWORK PROGRAMMING AND SIMULATION LAB [0 0 3 1]**

**Objectives:**

* To learn client server socket programming.
* To learn network concepts using packet tracer.

**Abstract:**

Socket programming, Simulation of Ethernet LAN protocol, token bus and token ring protocols, implementation of distance vector algorithm, link state routing algorithm and finding shortest path using packet tracer.

**Course Outcome:**

The students will be able to

* Implement client server socket programming.
* Simulate the network concepts using packet tracer.

**Reference:**

1. Stevens R., Stephen A. R., *Advanced Programming in the UNIX Environment* *(2e),* Pearson Education, 2013.
2. [Jesin A](https://www.flipkart.com/author/jesin-a), *Packet Tracer Network Simulator* *(1e),* Packt Publishing, 2014.

**SIXTH SEMESTER**

**HUM 3051: ENGINEERING ECONOMICS AND FINANCIAL MANAGEMENT**

**[2103]**

**Course Outcomes:**

On completion of this course the student should be able to:

* Apply the appropriate engineering economics analysis method(s) for problem solving
* Compute the depreciation of an asset using standard depreciation techniques
* Describe and apply the basic techniques of financial statement analysis
* Apply all mathematical approach models covered in solving engineering economics problems
* Analyse the responsibility of an engineer on risk and safety

**Abstract: Yet to be received**

**Time Value of money**

Time Value of Money, Interest Factors for Discrete Compounding, Nominal & Effective Interest Rates, Present and future worth of Single, Uniform, and Gradient cash flow. Related problems and case studies. [09]

**Economic Evaluation of Alternatives**

Bases for Comparison of Alternatives, Present worth amount, Capitalized Equivalent Amount, Annual Equivalent Amount, Future Worth Amount, Capital Recovery with Return, Rate of Return Method, Incremental Approach for Economic Analysis of Alternatives, Replacement analysis. Break Even Analysis for Single Product and Multi Product Firms, Break Even Analysis for Evaluation of Investment Alternatives. Minimum Cost Analysis.

[17]

# **Depreciation**

Physical & Functional Depreciation, Methods of Depreciation - Straight Line, Declining Balance, Double-Declining balance method, Sum-of-the-Years Digits, Sinking Fund and Service Output Methods, Case Study.

[02]

# **Financial Statement Analysis**

Balance Sheet and Profit & Loss Statement, Meaning & Contents. Ratio Analysis, Financial Ratios such as Liquidity Ratios, Leverage Ratios, Turn over Ratios, and Profitability Ratios, Drawbacks of Financial Statement Analysis. [06]

# **Project Risk**

Safety and Risk, Assessment of Risk and Safety, Case study, Risk Benefit Analysis and Reducing Risk [02]

# **References:**

1. Chan S. Park, "Contemporary Engineering Economics", 4th Edition, Pearson Prentice Hall, 2007.Michael L Shuler and Fikret Kargi. Bioprocess Engineering: Basic Concepts.
2. Thuesen G. J, "Engineering Economics", Prentice Hall of India, New Delhi, 2005.
3. Blank Leland T. and Tarquin Anthony J., "Engineering Economy", McGraw Hill, Delhi, 2002.
4. Prasanna Chandra, "Fundamentals of Financial Management", Tata McGraw Hill, Delhi, 2006.
5. Mike W. Martin and Roland Schinzinger, "Ethics in Engineering", Tata McGraw Hill, New Delhi, 2003.
6. Govindarajan M, Natarajan S, Senthil Kumar V. S, "Engineering Ethics", Prentice Hall of India, New Delhi,2004
7. Charles B. Fleddermann, "Engineering Ethics", Pearson Prentice Hall, New Jersey, 2012.

2013.

## **ICT 3253: DATA WAREHOUSING AND DATA MINING [3 0 0 3]**

**Objectives:**

* To understand the concepts of pre-processing to ease the mining process
* To learn data warehouse architecture and OLAP operations
* To gain insight into the data mining algorithms
* To study about outliers and detection of anomalies
* To learn basics of web mining

**Abstract:**

Introduction to Data mining- Technologies, applications, Major issues in data mining, Data objects & attribute types, Statistical descriptions of data, Data visualization, Data pre-processing-data cleaning, data integration, data transformation, data reduction, data discretization, Data warehouse modeling, Data warehouse design and usage, Data warehouse implementation, Association rule mining techniques- Apriori algorithm, Partition algorithm, Pincer Search algorithm, FP Tree Growth algorithm, PC tree algorithm, Dynamic Itemset Counting algorithm, Multilevel association rules, Classification and prediction techniques- Decision Tree Induction, Bayes Classification Methods, Techniques to Improve Classification Accuracy, Clustering techniques- Partitioning Methods, Hierarchical Methods, Density-Based Methods, Outlier detection, Web mining.

**Introduction**: Data Mining-Introduction, Kinds of data mined, Technologies used, targeted Applications and Major issues in Data mining. **[1 hour]**

**Data exploration**: Data objects & attribute types, Statistical descriptions of data, Data Visualization, Measuring data similarity & dissimilarity **[3 hours]**

**Data pre-processing**: Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization. **[6 hours]**

**Data warehouse**: Data warehouse definition, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, Data Generalization by Attribute-Oriented Induction **[5 hours]**

**Association rule mining**: Market Basket Analysis, Frequent Itemset Mining Methods-Apriori Algorithm, Partition algorithm, Pincer-Search algorithm, Dynamic Itemset Counting algorithm, Frequent pattern tree, PC tree, Mining Frequent Itemsets Using the Vertical Data Format, Pattern Evaluation Methods **[8 hours]**

**Classification and prediction**: Classification: Basic concepts, Decision Tree Induction, Bayes Classification Methods, Rule-Based Classification, Model Evaluation and Selection, Techniques to Improve Classification Accuracy **[5 hours]**

**Clustering**: Cluster analysis, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Evaluation of Clustering. **[4 hours]**

**Outlier detection**: Outliers and Outlier Analysis, Outlier Detection Methods, Clustering-Based Approaches, Classification-Based Approaches **[2 hours]**

**Web mining**: Overview of information retrieval and Text and Web Page Pre-Processing **[2 hours]**

**Outcomes:**

The students are able to

* Apply pre-processing techniques on datasets
* Describe the data warehouse architecture for facilitating querying
* Identify data mining techniques and apply on datasets
* Identify web mining techniques

**References:**

1. Han J. and Kamber M., *Data Mining: Concepts and Techniques (3e),* Morgan Kaufmann Publishers, 2011.
2. Pujari A. K., *Data Mining Techniques* *(4e),* Orient Blackswan, 2016.
3. Pang-N. T., Steinbach M., Anuj K., Vipin K., *Introduction to Data Mining, Pearson Education* *(2e)*, Pearson 2018.
4. Bing L., *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (2e),* Springer, Second Edition, 2011.

**ICT 3254: DISTRIBUTED SYSTEMS [3 0 0 3]**

**Objectives:**

* To learn the features of distributed systems
* To learn the communication techniques in distributed systems.
* To understand the file system, name services, shared memory pertaining to the distributed systems
* To understand the concepts of coordination, consistency, fault tolerance in the distributed environment

**Abstract:**

Introduction, Distributed Systems Models, Challenges, Processes and Communication, External data representation, Case study - Unix inter process communication, Distributed objects, Remote procedure call, Case study: Java RMI, SUN RPC, Distributed File System, Naming Systems, Directory and discovery services, Case study: SUN NFS / GNS, Distributed Shared Memory, DSM implementation algorithms, Time and Global States, Logical and vector clocks, Coordination and agreement, Distributed transactions, Consistency and Replication Management, Fault Tolerance, Case Study - Google.

**Syllabus:**

**Characterization of Distributed Systems and System Models:**

Introduction, Examples of Distributed Systems, Focus on resource sharing, Challenges, Architectural models, Distributed system design issues **[4 Hours**]

**Processes and Communication:**

Introduction, Client and server process design and issues, External data representation and marshalling. Communication types/patterns, Case study - Unix inter process communication. **[4 Hours]**

**Distributed objects and Remote Invocation:**

Communication between distributed objects, Remote procedure call. Case study: Java RMI, SUN RPC. **[3 Hours]**

**Distributed File systems and Name Services:**

Introduction, File service architecture, Recent advances, Name services and the domain name system, Directory and discovery services. Case study: SUN NFS / GNS **[6 Hours]**

**Distributed Shared Memory:**

Introduction, Advantages, Disadvantages, DSM implementation algorithms **[2 Hours]**

**Time and Global States, Coordination, and agreement:**

Clocks, logical clock and vector clocks, global states, Distributed mutual exclusion, Election algorithms, Global positioning of nodes. **[6 Hours]**

**Distributed Transactions, Consistency, and Replication:**

Introduction, Distributed Transactions and Concurrency Control, Consistency models- Data- centric and client centric models, Replica management, consistency management protocols.

**[6 Hours]**

**Fault Tolerance:** Introduction, process resilience, reliable Client-server communication.

**[3 Hours]**

**Case Study for Designing Distributed Systems**: Google. **[2 Hours]**

**Outcomes:**

**Upon completion of the course, the students will be able to:**

* Understand basic concepts of distributed systems.
* Understand various techniques used for communication between distributed systems
* Analyze the requirements for designing distributed file system and shared memory
* Understand the issues pertaining to coordination, consistency, fault tolerance for a distributed system
* Identify the concepts of distributed system to design any distributed application

**References:**

1. George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, *Distributed Systems Concepts and Design (5e),* Pearson Education, 2017.
2. Andrew Tanenbaum S., *Distributed Systems: Principles and Paradigms (2e),* Pearson Education Asia, 2016.
3. Singhal M., Shivaratri N., *Advanced Concepts in Operating Systems (2e),* McGraw Hill Education, 2017.
4. Andrew Tanenbaum S., *Modern Operating System (3e),* Pearson Education International, 2016.

**ICT \*\*\*\*: PROGRAM ELECTIVE-I [3 0 0 3]**

**ICT \*\*\*\*: PROGRAM ELECTIVE-II [3 0 0 3]**

**\*\*\* \*\*\*\*: OPEN ELECTIVE-III**

**ICT 3264: ADVANCED TECHNOLOGY LAB [0 0 3 1]**

**Objectives:**

* To recognize the emerging technology used in software industry
* To illustrate the graphical user interface(GUI) design for the application.
* To choose the database system that interacts with the application.
* To demonstrate project life cycle management.

**Abstract:**

Introduction to the emerging technology (android, react native, etc.) used in the software industries, Understand the tools and techniques used to application development, Application user interface design, Backend database design, Implementation of mini project.

**Outcome:**

After completing the course the students will be able to:

* Recognize the emerging technology used in the software industry.
* Construct the database that interacts with the application.
* Develop the GUI for the application.
* Select the appropriate tools and techniques to implement application.

**References**

1. McConnell S., *Code Complete: A Practical Handbook of Software Construction (2e)*, Microsoft Press, Re-print 2015.
2. Horton J., *Android Programming for Beginners,* Packt Publishing Ltd, 2015.
3. Griffiths D. , *Head First Android Development (1e)*, O’Reilly Media, Inc., 2015.
4. Holmes E. and Bray T., *Getting Started with React Native,* Packt Publishing Ltd, 2015.

## **ICT 3265: DATA WAREHOUSING AND DATA MINING LAB [0 1 2 2]**

**Objectives:**

* To get furnished with pre-processing tools.
* To gain practical key skills in developing data warehouse.
* To apply various data mining algorithms on raw datasets.
* To obtain hands-on experience of providing data mining solutions to real world problems.

**Abstract:**

Data Pre-processing- Cleaning, Integration, Transformation, Reduction, Implementation of data warehouse on pre-processed data- Creating Physical data model, Creating data flow based on physical data model, Creating control flow- Sequential and parallel, Cube creation, Data mining solutions for real world problems- Implementing association algorithm, classification algorithm, clustering algorithm, Development of mini project.

**Outcomes:**

The students are able to:

* Identify suitable pre-processing techniques for various dataset.
* Develop data warehouse.
* Apply suitable data mining technique on preprocessed data.

**References:**

1. Han J. and Kamber M., *Data Mining: Concepts and Techniques* *(3e),* Morgan Kaufmann Publishers, 2011.
2. Pujari A. K., *Data Mining Techniques* *(4e),* Orient Blackswan, 2016.
3. Silberschatz A., Korth H. F., Sudarshan S., *Database System Concepts* *(6e),* McGraw Hill Education, 2013.

## **ICT 3266: INTERNET TOOLS AND TECHNOLOGY LAB [0 1 2 2]**

**Objectives**

* To design static and dynamic web pages.
* To gain familiarity in Client and Server-Side Programming using latest Internet Tools and Technologies.
* To learn Database Connectivity for web applications

**Abstract:**

HTML, XHTML, Java Script, Introduction to Python Programming, Advanced Python programming concepts that include applications involving connection to databases; CGI Programming with Python; JSON and JQuery; NodeJS for client-side programming; Angular JS for server side programming; mini project.

**Outcomes**

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

* Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
* Have a good understanding of Web Application Terminologies, Internet Tools other web services.
* Understand and implement the client and server side programming for a Web application using latest tools.

**References:**

1. Deitel H. M., Deitel P.J. & Goldberg A.B., *Internet & World Wide Web How To Program (5e),* Pearson Education, 2011.
2. Bates C., *Web Programming: Building Internet Application (3e)*, Wiley India, 2012.

**SEVENTH SEMESTER**

There are five program electives and one open elective with total of 18 credits to be taught in this semester.

**ICT \*\*\*\*: PROGRAM ELECTIVE-III [3 0 0 3]**

**ICT \*\*\*\*: PROGRAM ELECTIVE-IV [3 0 0 3]**

**ICT \*\*\*\*: PROGRAM ELECTIVE-V [3 0 0 3]**

**ICT \*\*\*\*: PROGRAM ELECTIVE-VI [3 0 0 3]**

**ICT \*\*\*\*: PROGRAM ELECTIVE-VII [3 0 0 3]**

**\*\*\* \*\*\*\* OPEN ELECTIVE IV [3]**

**EIGHTH SEMESTER**

**ICT 4298: INDUSTRIAL TRAINING**

Each student has to undergo industrial training for a minimum period of 4 weeks. This may be taken in a phased manner during the vacation starting from the end of third semester. Student has to submit to the department a training report in the prescribed format and also make a presentation of the same. The report should include the certificates issued by the industry.

**ICT 4299: PROJECT WORK/PRACTICE SCHOOL**

The project work may be carried out in the institution/industry/ research laboratory or any other competent institutions. The duration of the project work shall be a minimum of 16 weeks which may be extended up to 24 weeks. A mid-semester evaluation of the project work shall be done after about 8 weeks. An interim project report on the progress of the work shall be submitted to the department during the mid-semester evaluation. The final evaluation and viva-voice will be conducted after submission of the final project report in the prescribed form. Student has to make a presentation on the work carried out, before the department committee as part of project evaluation.

**MINOR SPECIALIZATION**

**I.COMPUTATIONAL INTELLIGENCE**

**CSE 4053: ARTIFICIAL INTELLIGENCE [3 0 0 3]**

**Objectives:**

* To represent the real world problem and devise mechanisms to reach the goal description
* To analyse hypothetical game trees to predict the best move for each player in a multiplayer zero sum games
* Infer the necessary axioms from the existing knowledge base in Propositional and Predicate logic.
* To understand representation of knowledge, uncertainties and reasoning

**Abstract:**

Foundations of Artificial Intelligence, History of Artificial Intelligence, The state of the Art. Agents and Environments, The concept of Rationality, The Nature of Environments, The structure of Agents. Problem Solving agents, Example Problems, Searching for Solutions, Uninformed search strategies, informed (Heuristic) search strategies, Heuristic functions. Games, Optimal decision in games, Alpha Beta Pruning Knowledge based agents, The Wumpus World, Logic, Propositional logic, Propositional Theorem Proving. Representation revisited, Syntax and semantics of First order logic, Using First order logic, Knowledge engineering in First order. Ontological Engineering, Categories and objects, Reasoning systems for categories, The internet shopping world. Acting under uncertainty, Basic probability notation, Baye’s rule, representing knowledge in uncertainties, semantics of Bayesian networks.

**Syllabus:**

**INTRODUCTION :**

What is AI? Foundations of Artificial Intelligence, History of Artificial Intelligence, The state of the Art **[03 Hours]**

**Intelligent Agents:**

Agents and Environments, The concept of Rationality, The Nature of Environments, The structure of Agents. **[06 Hours]**

**Solving Problems by Searching**:

Problem Solving agents, Example Problems, Searching for Solutions, Uninformed search strategies, Informed (Heuristic) search strategies, Heuristic functions. **[8 Hours]**

**Adversarial Search**:

Games, Optimal decision in games, Alpha Beta Pruning **[5 Hours]**

**Logical Agents:**

Knowledge based agents, The Wumpus World, Logic, Propositional logic, Propositional Theorem Proving.Representation revisited, Syntax and semantics of First order logic, Using First order logic, Knowledge engineering in First order **[8 Hours]**

**Knowledge Representation**:

Ontological Engineering, Categories and objects, Reasoning systems for categories, The internet shopping world. **[6 Hours]**

**QUANTIFYING UNCERTAINITY AND PROBABILISTIC REASONING**

Acting under uncertainty, Basic probability notation, Baye’s rule, representing knowledge in uncertainties, semantics of Bayesian networks.

**Outcomes:**

After studying this course, students will be able to:

* Understand concepts of rational agents and represent real world problems.
* Devise mechanisms to reach the goal state using searching techniques
* Analyze hypothetical game trees and knowledge representation
* Infer the necessary axioms from the existing knowledge base in logic.
* Understand basics of uncertainty, expert systems and machine learning.

**References:**

1. Russell S., and Norvig P., *Artificial Intelligence A Modern Approach* (3e), Pearson 2010.
2. Rich E., Knight K., Nair S.B., *Artificial Intelligence* (3e), Tata McGraw Hill, 2008.

**ICT 4031:** **COMPUTER VISION [3 0 0 3]**

**Objectives:**

* Learn formation of image and different camera models and their calibration.
* Learn different feature detection and mathematics of description methods.
* Learn object recognition through different learning algorithms.

**Abstract:**

Introduction to computer vision and its applications, Image formation, Liner Filtering, magetransformations and Colour models, Edge Detection methods (Laplacian detectors and Canny edge detector), Points and patches, Harris corner detector, Histogram of Gradients, Difference of Gaussian detector, SIFT, Colour and Texture, Feature based alignment, least squares and RANSAC, Camera models, Camera calibration, Stereo vision, Stereo correspondence, Epipolar geometry, Optical flow, Lucas Kanade method, KLT tracking method, Mean shift method, Dense motion estimation, Support Vector Machines, Face detection and recognition, Bag of words, Deep learning.

**Syllabus:**

**INTRODUCTION:**

Introduction to computer vision and its applications, Image formation, Liner Filtering, Image transformations and Colour models. [**07 Hours]**

**FEATURE DETECTION AND MATCHING:**

Edge Detection methods (Laplacian detectors and Canny edge detector)**,** Points and patches, Harris corner detector, Histogram of Gradients, Difference of Gaussian detector, SIFT, Colour and Texture, Feature based alignment, least squares and RANSAC [**09 Hours]**

**CAMERA CALIBRATION:**

Camera models, Camera calibration, Stereo vision, Stereo correspondence, Epipolar geometry

**[08 Hours]**

**TRACKING:**

Optical flow, Lucas Kanade method, KLT tracking method, Mean shift method, Dense motion estimation. **[06 Hours]**

**OBJECT RECOGNITION**:

Support Vector Machines, Face detection and recognition, Bag of words, Deep learning.

**[06 Hours]**

**Outcomes:**

After studying this course, students will be able to:

* Understand the concepts of image formation, colour models and linear filtering.
* Understand the mathematics behind feature detection and description methods.
* Demonstrate a thorough understanding of fundamental concepts in camera calibration.
* Understand and analyze various object tracking algorithms.
* Comprehend object and scene recognition and categorization from images.

**References:**

1. Szeliski R., *Computer Vision: Algorithms and Applications*, Springer 2011.
2. David A. F. and Ponce J., *Computer Vision: A Modern Approach, PHI learning* 2009.
3. Solem J. E., *Programming Computer Vision with Python*, O’Reilly, 2012.

**ICT 4032: MACHINE LEARNING [3 0 0 3]**

**Objectives:**

This course will enable students to

* Gain basic knowledge about the key algorithms and theory that forms the foundation of machine learning and computational intelligence
* Get a practical knowledge of machine learning algorithms and methods

**Abstract:**

Introduction to Machine Learning, Mathematical Preliminaries, Supervised Learning-LMS, logistic regression, GDA, Naive Bayes, SVM, model selection, Learning theory-bias/variance tradeoff, union and Chernoff bounds, VC dimensions, Unsupervised learning-clustering, k-means, Gaussian mixture, factor analysis, PCA, ICA, Reinforcement learning-MDPs, Bellman equations, value and policy iteration, LQR, LQG, Q-learning, policy search, POMDPs

**Syllabus:**

**Introduction:**

Machine learning basics, Examples of Machine learning application, Steps in developing machine learning application [**3 Hour**]

**CLASSIFICATION AND REGRESSION:**

Bayesian Decision Theory: Continuous features, Minimum Error Rate Classification, Classifiers, Discriminate Functions, Error Probabilities and Integral, Bayesian Belief Network, Maximum likelihood ratio, parametric classification, regression, Multivariate methods, Naïve Bayesian Model. Non-Parametric Techniques: Density Estimation, Parzen Windows, kn- Nearest-Neighbor Estimation, Nearest – Neighbor Rule, K- nearest neighbor classification.  **[14 Hours]**

**SUPERVISED LEARNING:**

Linear discrimination, Gradient descent, Logistic discrimination, Single layer Perceptron, Training a perceptron, Multilayer perceptron, Back-Propagation Algorithm, Genetic algorithms, Decision trees, Support vector machines. **[12 Hours]**

**UNSUPERVISED LEARNING:**

Clustering, K-Means clustering, EM-algorithm, Hierarchical clustering, Competitive learning, and Radial basis functions. [**5 Hours]**

**COMBINING MULTIPLE LEARNERS:**

Voting, Error correcting output codes, Bagging, Boosting. [**2 Hours]**

**Outcomes:**

* Understand the principles, advantages, limitations and possible applications of machine learning.
* Identify and apply the appropriate machine learning technique for classification, using pattern recognition system.
* Analyze simple algorithms for supervised learning, reinforcement learning, and unsupervised learning.
* Understand unsupervised learning for data clustering.
* Validation of a combined machine learning system.

**References:**

1. Murphy K.P., *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.
2. Mohri M., Rostamizadeh A., and Talwalkar A., *Foundations of Machine Learning*, MIT Press, 2012.
3. Koller D., and Friedman N, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, 2009.
4. Bishop C.M., *Pattern Recognition and Machine Learning* (2e), Springer, 2013.

**CSE 4054: SOFT COMPUTING PARADIGMS [3 0 0 3]**

**Course objectives:**

This course will enable students to

* To Conceptualize the working of human brain using ANN.
* To become familiar with neural networks that can learn from available examples and generalize to form appropriate rules for inference systems.
* To introduce the ideas of fuzzy sets, fuzzy logic and use of heuristics based on human experience.
* To provide the mathematical background for carrying out the optimization and familiarizing genetic algorithm for seeking global optimum in self-learning situation.

**Abstract:**

Soft Computing, Artificial Intelligence, Soft-Computing Techniques, Expert Systems Types of Problems, Modeling the Problem, Machine Learning, Handling Impreciseness, Clustering , Hazards of Soft Computing, Road Map for the Future Artificial Neural Networks, The Biological Neuron, The Artificial Neuron, Multilayer Perceptron, Modeling the Problem, Types of Data Involved, Training, Issues in ANN, Example of Time Series Forecasting Types of Artificial Neural Networks, Radial Basis Function Network, Learning Vector Quantization, Self-Organizing Maps, Recurrent Neural Network, Hopfield Neural Network, Adaptive Resonance Theory, Character Recognition by Commonly Used ANNs Fuzzy Systems, Fuzzy Logic, Membership Functions, Fuzzy Logical Operators, More Operations, Fuzzy Inference Systems, Type-2 Fuzzy systems, Other Sets, Sugeno Fuzzy Systems, Example: Fuzzy Controller Evolutionary Algorithms: Evolutionary Algorithms, Biological Inspiration Evolutionary Algorithms Genetic Algorithms, Fitness Scaling, Selection, Mutation, Crossover, Other Genetic Operators, Algorithm Working, Diversity, Grammatical Evolution, Other Optimization Techniques, Metaheuristic Search, Traveling Salesman Problem Introduction, Key Takeaways from Individual Systems, Adaptive Neuro-Fuzzy Inference Systems, Evolutionary Neural Networks, Evolving Fuzzy Logic, Fuzzy Artificial Neural Networks with Fuzzy Inputs, Rule Extraction from ANN, Modular Neural Network

**Syllabus:**

**INTRODUCTION** :

Soft Computing, Artificial Intelligence, Soft-Computing Techniques, Expert Systems Types of Problems, Modeling the Problem, Machine Learning, Handling Impreciseness, Clustering , Hazards of Soft Computing, Road Map for the Future [**10 Hours]**

**ARTIFICIAL NEURAL NETWORK –I** :

Artificial Neural Networks, The Biological Neuron, The Artificial Neuron, Multilayer Perceptron, Modeling the Problem, Types of Data Involved, Training, Issues in ANN, Example of Time Series Forecasting

**ARTIFICIAL NEURAL NETWORKS II :**

Types of Artificial Neural Networks, Radial Basis Function Network, Learning Vector Quantization, Self-Organizing Maps, Recurrent Neural Network, Hopfield Neural Network, Adaptive Resonance Theory, Character Recognition by Commonly Used ANNs **[10 Hours]**

**FUZZY INFERENCE SYSTEMS** :

Fuzzy Systems, Fuzzy Logic, Membership Functions, Fuzzy Logical Operators, More Operations, Fuzzy Inference Systems, Type-2 Fuzzy systems, Other Sets, Sugeno Fuzzy Systems, Example: Fuzzy Controller Evolutionary Algorithms: Evolutionary Algorithms, Biological Inspiration

**EVOLUTIONARY ALGORITHMS**:

Evolutionary Algorithms Genetic Algorithms, Fitness Scaling, Selection, Mutation, Crossover, Other Genetic Operators, Algorithm Working, Diversity, Grammatical Evolution, Other Optimization Techniques, Metaheuristic Search, Traveling Salesman Problem [**10 Hours]**

**HYBRID SYSTEMS**:

Introduction, Key Takeaways from Individual Systems, Adaptive Neuro-Fuzzy Inference Systems, Evolutionary Neural Networks, Evolving Fuzzy Logic, Fuzzy Artificial Neural Networks with Fuzzy Inputs, Rule Extraction from ANN, Modular Neural Network [**06 Hours]**

**Course outcomes:**

Learner will be able to…

* Ability to analyze and appreciate the applications which can use fuzzy logic.
* Ability to design inference systems.
* Ability to understand the difference between learning and programming and explore practical applications of Neural Networks (NN).
* Ability to appreciate the importance of optimizations and its use in computer engineering fields and other domains.
* Students would understand the efficiency of a hybrid system and how Neural Network and fuzzy logic can be hybridized to form a Neuro-fuzzy network and its various applications.

**Reference Books:**

1. Shukla A., Tiwari R., Kala R., *Real Life Applications of Soft Computing*, CRC Press, Taylor and Francis Group, London 2010.
2. Ross T.J., *Fuzzy Logic with Engineering Applications*, Wiley publication, 2010.
3. Sivanandam S.N., Deepa S.N., *Principles of Soft Computing*, (2e), Wiley Publication, 2010.
4. Rajasekaran S., and Pai G. A. V., *Neural Networks, Fuzzy Logic and Genetic Algorithms*, PHI Learning, 2010.
5. Jang J. S .R., *Neuro-Fuzzy and Soft Computing*, PHI 2003.

**II. COMPUTER GRAPHICS AND VISUALIZATION**

**CSE 4051: AUGMENTED AND VIRTUAL REALITY [3 0 0 3]**

**Course objectives:**

This course provides students with an opportunity to explore the research issues in Augmented Reality and Virtual Reality (AR &VR).

* To give basic understanding and the big picture of the principles, methods and applications of VR, AR and related
* It also makes the students know the basic concept and framework of virtual reality.

**Abstract:**

Introduction of Virtual and Augmented reality, Definition and scope, A Brief History 3 I’s of Virtual Reality A Short History of Early Virtual Reality, Early Commercial VR Technology, VR Becomes an Industry, Components of a VR System, Displays In augmented reality Multimodal Displays, Audio Displays Haptic, Tactile, and Tangible Displays, Displays, Visual Perception, Requirements and Characteristics, Multiple Models of I/O Gesture Interfaces Three-Dimensional Position Trackers, Navigation and Manipulation Interfaces, Gesture Interfaces. Output Devices, Haptic Display, Graphics Displays, Sound Displays. Computer Vision for Augmented Reality, Natural Feature Tracking by Detection, Incremental Tracking, Simultaneous Localization and Mapping, Outdoor Tracking, Computing Architectures for VR, The Rendering Pipeline, Workstation-Based Architectures, Distributed VR Architectures, Geometric Modeling, Kinematics Modeling, Physical Modeling, Behavior Modeling.

**Syllabus:**

**INTRODUCTION OF AUGMENTED REALITY, DEFINITION AND SCOPE**

Definition and Scope, A Brief History of Augmented Reality, Examples **[06 Hours]**

Introduction The Three I’s of Virtual Reality, A Short History of Early Virtual Reality, Early Commercial VR Technology, VR Becomes an Industry, The five Classic Components of a VR System.

Displays in augmented reality, Multimodal Displays, Audio Displays Haptic, Tactile, and Tangible Displays, Ol factory and Gustatory Displays, Visual Perception, Requirements and Characteristics.

**[10 Hours]**

Multiple Models of Input and Output Interface in Virtual Reality: Gesture Interfaces Three-Dimensional Position Trackers, Navigation and Manipulation Interfaces Gesture Interfaces. Output Devices: Graphics, Three-Dimensional Sound, and Haptic Display , Graphics Displays, Sound Displays ,Haptic Feedback

**COMPUTER VISION FOR AUGMENTED REALITY :**

Marker Tracking, Multiple-Camera Infrared Tracking, Natural Feature Tracking by Detection, Incremental Tracking, Simultaneous Localization and Mapping, Outdoor Tracking [**10 Hours]**

**COMPUTING ARCHITECTURES FOR VR:**

The Rendering Pipeline, PC Graphics Architecture, Workstation-Based Architectures, Distributed VR Architectures

**MODELING:**

Geometric Modeling, Kinematics Modeling, Physical Modeling, Behavior Modeling, Model Management [**10 Hours]**

**Course outcomes:**

After studying this course, students will be able to:

* Understand Fundamentals of virtual reality systems, including geometric modeling, transformations, graphical rendering.
* Understand the technology for managing large scale VR environment in real time.
* Learn technology for multimodal user interaction and perception in VR, in particular the visual, audial and haptic interface and behavior
* To teach students the technology for managing large scale VR environment in real time.
* Apply the gained knowledge in developing VR system framework and development tools.

**References:**

1. Burdea, G. C. and P. Coffet. *Virtual Reality Technology (2e),* Wiley-IEEE Press, 2006.
2. Dieter Schmalstieg, Tobias Hollerer, *Augmented Reality: Principles & Practice (1e),* Addison-Wesley, 2016.
3. Tony parisi, *Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile (1e),* O’Reilly Media, 2015.
4. Steve Aukstakalnis, *Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR (Usability) (1e),* Addison-Wesley Professional, 2016.
5. Jonathan Linowes, *Unity Virtual Reality Projects* *Paperback*, Packt Publishing eBooks Account, September 2015.

**ICT 4033: COMPUTER GRAPHICS [3 0 0 3]**

**Pre requisites:**

* A basic course on Mathematics: Coordinate geometry and transformation; Basic linear algebra such as matrix multiplication.

**Course Objectives:**

* Understand the basics of Graphics Architecture, Output Primitives algorithms,
* Understand the Geometrical Transformations of 2D/3D, Solid Modeling.
* Understand the importance of Dialogue Design, Human machine Interaction and virtualism
* Comprehend and analyze the fundamentals of light, animation and visible surface detection.

**Abstract:**

Introduction: History of computer graphics and applications, Introduction to OpenGL, Geometric

transformations: Homogeneous coordinates, affine transformations (translation, rotation, scaling,

shear, reflection), concatenation, matrix stacks and use of model view matrix in OpenGL for these

operations Examples, Viewing (3D), Visibility- z-Buffer, BSP trees, Open-GL culling, hiddensurface algorithms, Shading, Rasterization- Line segment and polygon clipping, 3D clipping, scan conversion, polygonal fill, Bresenham's algorithm, Discrete Techniques: Texture mapping,compositing, textures in OpenGL; Ray Tracing, Representation and Visualization: Bezier curves and surfaces, B-splines, visualization, interpolation, marching squares algorithm.

**Syllabus:**

**INTRODUCTION :**

History of computer graphics, applications, graphics pipeline, physical and synthetic images, synthetic camera, modeling, animation, rendering, relation to computer vision and image processing, review of basic mathematical objects (points, vectors, matrix methods) [**8 Hours]**

**INTRODUCTION TO OPENGL :**

OpenGL architecture, primitives and attributes, simple modeling and rendering of two- and three-dimensional geometric objects, indexed and RGB color models, frame buffer, double buffering, GLUT, interaction, events and callbacks, picking

**GEOMETRIC TRANSFORMATIONS :**

Homogeneous coordinates, affine transformations (translation, rotation, scaling, shear, reflection), concatenation, matrix stacks and use of model view matrix in OpenGL for these operations Examples [**8 Hours]**

**VIEWING :**

Classical three dimensional viewing, computer viewing, specifying views, parallel and perspective projective transformations; Visibility- z-Buffer, BSP trees, Open-GL culling, hidden-surface algorithms [**7 Hours]**

**SHADING :**

Light sources, illumination model, Gouraud and Phong shading for polygons. Rasterization- Line segment and polygon clipping, 3D clipping, scan conversion, polygonal fill, Bresenham's algorithm [**9 Hours]**

**DISCRETE TECHNIQUES:**

Texture mapping, compositing, textures in OpenGL; Ray Tracing- Recursive ray tracer, ray-sphere intersection

**REPRESENTATION AND VISUALIZATION :**

Bezier curves and surfaces, B-splines, visualization, interpolation, marching squares algorithm

**[4 Hours]**

**Course outcomes:**

After learning the course the students should be able to:

* Apply the fundamental concepts within computer graphics such as geometrical transformations, illumination models, removal of hidden surfaces and rendering
* Design the ideas in some fundamental algorithms for computer graphics and to some extent be able to compare and evaluate them
* Implement and apply fundamental principles within interaction programming
* Implement and understand fundamental concepts within information visualization and scientific visualization.

**Reference Books:**

1. Hearn D and Baker P., *Computer Graphics with OpenGL (4e),* Pearson, 2014.
2. Edward Angel, *Interactive Computer Graphics A Top-Down Approach Using OpenGL* (*6e),* Pearson Education, 2011.
3. F. S. Hill Jr. and S. M. Kelley, *Computer Graphics using OpenGL (3e),* Prentice Hall, 2006.
4. Peter Shirley and Steve Marschner, *Computer Graphics (1e),* A. K. Peters, 2010.

**ICT 4031: COMPUTER VISION [3 0 0 3]**

**Objectives:**

* Learn formation of image and different camera models and their calibration.
* Learn different feature detection and mathematics of description methods.
* Learn object recognition through different learning algorithms.

**Abstract:**

Introduction to computer vision and its applications, Image formation, Liner Filtering, magetransformations and Colour models, Edge Detection methods (Laplacian detectors and Canny edge detector), Points and patches, Harris corner detector, Histogram of Gradients, Difference of Gaussian detector, SIFT, Colour and Texture, Feature based alignment, least squares and RANSAC, Camera models, Camera calibration, Stereo vision, Stereo correspondence, Epipolar geometry, Optical flow, Lucas Kanade method, KLT tracking method, Mean shift method, Dense motion estimation, Support Vector Machines, Face detection and recognition, Bag of words, Deep learning.

**Syllabus:**

**INTRODUCTION:**

Introduction to computer vision and its applications, Image formation, Liner Filtering, Image transformations and Colour models. [**07 Hours]**

**FEATURE DETECTION AND MATCHING:**

Edge Detection methods (Laplacian detectors and Canny edge detector)**,** Points and patches, Harris corner detector, Histogram of Gradients, Difference of Gaussian detector, SIFT, Colour and Texture, Feature based alignment, least squares and RANSAC [**09 Hours]**

**CAMERA CALIBRATION:**

Camera models, Camera calibration, Stereo vision, Stereo correspondence, Epipolar geometry

**[08 Hours]**

**TRACKING:**

Optical flow, Lucas Kanade method, KLT tracking method, Mean shift method, Dense motion estimation. **[06 Hours]**

**OBJECT RECOGNITION**:

Support Vector Machines, Face detection and recognition, Bag of words, Deep learning.

**[06 Hours]**

**Outcomes:**

After studying this course, students will be able to:

* Understand the concepts of image formation, colour models and linear filtering.
* Understand the mathematics behind feature detection and description methods.
* Demonstrate a thorough understanding of fundamental concepts in camera calibration.
* Understand and analyze various object tracking algorithms.
* Comprehend object and scene recognition and categorization from images.

**References:**

1. Szeliski R., *Computer Vision: Algorithms and Applications*, Springer 2011.
2. David A. F. and Ponce J., *Computer Vision: A Modern Approach, PHI learning* 2009.
3. Solem J. E., *Programming Computer Vision with Python*, O’Reilly, 2012.

**CSE 4052: DIGITAL IMAGE PROCESSING [3 0 0 3]**

**Course objectives:**

This course will enable students to

* Learn the fundamental concepts of a digital image processing system and process the image in spatial domain
* Analyse algorithms for image enhancement in frequency domain
* Identify the noise models and apply image restoration methods for noisy images
* Study the various morphological algorithms and implement them
* Analyse the various segmentation methods, and apply it for images

**Abstract:**

Introduction, components of image processing system, Spatial domain transformations, histogram processing, smoothing, sharpening spatial filters, Filtering in the frequency domain- Introduction to Fourier transform, image smoothing, image sharpening using frequency domain filters. Image restoration- Noise models, restoration using spatial filtering, periodic noise reduction by frequency domain filtering, Morphological image processing- Preliminaries, dilation and erosion, opening and closing, hit-or-miss transformation, basic algorithms, extension to gray-scale images, Image segmentation- Point, line, and edge detection, Thresholding, Region Segmentation Using Clustering and Superpixels, Graph Cuts, morphological watersheds, motion in segmentation.

**Syllabus:**

**INTRODUCTION AND IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN:**

Fundamental Steps in Digital Image Processing, Fields that use Digital Image Processing, Components of an Image Processing System, Elements of Visual Perception, Image Sampling and Quantization, Basic Relationships between Pixels [**8 Hours]**

Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic/Logic Operations, Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters, Combining Spatial Enhancement Methods.

**FILTERING IN THE FREQUENCY DOMAIN**:

Introduction to the Fourier Transform and the Frequency Domain, Sampling and the Fourier Transform of Sampled Functions, The Discrete Fourier Transform of One Variable, Extensions to Functions of Two Variables, Some Properties of the 2-D DFT and IDFT, The Basics of of Ffiltering in the Frequency Domain, Image Smoothing Using Frequency Domain Filters, Image Sharpening Using Frequency Domain [**6 Hours]**

**IMAGE RESTORATION**:

A Model of the Image Degradation/Restoration Process, Noise models, Restoration in the presence of noise only spatial filtering, Periodic noise reduction by frequency domain filtering, estimating the degradation Function. [**6 Hours]**

**IMAGE SEGMENTATION:**

Point, Line, and Edge Detection, Thresholding, Segmentation by Region Growing and by Region Splitting and Merging, Region Segmentation Using Clustering and Superpixels, Region Segmentation Using Graph Cuts, Segmentation Using Morphological Watersheds, Use of Motion in Segmentation [**8 Hours]**

**MORPHOLOGICAL IMAGE PROCESSING**:

Preliminaries, Dilation and Erosion, Opening and Closing, The hit-or-miss transformation, some basic Morphological algorithms, Gray-Scale Morphology(except Morhological Reconstruction)

**[8 Hours]**

**Course outcomes:**

After studying this course, students will be able to:

* Understand the fundamental concepts of a digital image processing system and process the image in spatial domain
* Understand and implement algorithms for image enhancement in frequency domain
* Identify the noise models and apply image restoration methods for noisy images
* Describe the various morphological algorithms and implement them
* Have knowledge of the various segmentation methods and apply it for various images

**Reference Books:**

1. Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing* *(4e),* Pearson, 2017.
2. Milan Sonka, Vaclav Hlavac, Roger Boyle, *Image Processing, Analysis and Machine Vision (4e*), CENGAGE Learning, 2014.
3. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, *Digital Image Processing Using MATLAB (2e),* Mc Graw Hill India, 2010.
4. Gloria Bueno García, Oscar Deniz Suarez, José Luis Espinosa Aranda, Jesus Salido Tercero, Ismael Serrano Gracia, Noelia Vállez Enano, Learning Image Processing with *OpenCV (1e),* Packt Publishing, 2015.

**III.DATA ANALYTICS**

**ICT 4034: BIG DATA ANALYTICS [3 0 0 3]**

**Objectives:**

* Understand concept and challenges of big data
* Use Hadoop and MapReduce framework
* Understand data analytics using Spark.
* Build and query NOSQL data stores for large-volume and semi-structured data

**Abstract:**

Understanding Big Data: Concepts and Terminology, Big Data Characteristics, Different Types of Data, Case Study Background. Big data and Hadoop: Understanding Hadoop features, Learning the HDFS and MapReduce architecture, Understanding Hadoop subprojects, Understanding the basics of MapReduce, Introducing Hadoop MapReduce, Understanding the Hadoop MapReduce fundamentals, Writing a Hadoop MapReduce example. Spark and Big Data: Theoretical concepts in Spark, Core components of Spark, The Spark architecture, Spark SQL, Spark Streaming. NoSQL databases: Need for NoSQL, NoSQL databases, In-memory databases, Columnar databases, Document-oriented databases, Key-value databases, Graph databases, Other NoSQL types and summary, Working on NoSQL systems using MongoDB. Applications: Implementation of machine learning algorithms using MapReduce and Spark.

Understanding Big Data: Concepts and Terminology, Big Data Characteristics, Different Types of Data, Case Study Background [Textbook1] **[4 Hours]**

Big data and Hadoop: Understanding Hadoop features, Learning the HDFS and MapReduce architecture, Understanding Hadoop subprojects, Understanding the basics of MapReduce, Introducing Hadoop MapReduce, Understanding the Hadoop MapReduce fundamentals, Writing a Hadoop MapReduce example [Textbook 2, 4] **[12 Hours]**

Spark and Big data: Theoretical concepts in Spark, Core components of Spark, The Spark architecture, Spark SQL, Spark Streaming [Textbook 3, 5] **[10 Hours]**

NOSQL: Need for NoSQL, NoSQL databases, In-memory databases, Columnar databases, Document-oriented databases, Key-value databases, Graph databases, Other NoSQL types and summary, Working on NoSQL systems using MongoDB [Textbook 3] **[4 Hours]**

Module 5: Implementation of machine learning algorithms using MapReduce and Spark **[6 Hours]**

**Course Outcomes:**

At the end of the course of studies, students will have acquired the ability to

* Explain Hadoop Architecture
* Design Big data solutions using MapReduce framework.
* Design Big data solutions using Spark
* Model large, semi-structured data using NOSQL .

**References:**

1. Erl T., Khattak W., and Buhler P., *Big Data Fundamentals, Concepts, Drivers & Techniques* *(1e)*, The Prentice Hall Service Technology Series, 2016.
2. Prajapati V., *Big Data Analytics with R and Hadoop*, Packt Publishing Ltd., 2013.
3. Dasgupta N., *Practical Big Data Analytics*, Packt Publishing Ltd, 2018.
4. Rajaraman A., and Ullman J. D., *Mining of Massive Datasets*, Cambridge University Press, 2011.
5. Zaharia [M.](https://www.safaribooksonline.com/search/?query=author%3A%22Matei%20Zaharia%22&sort=relevance&highlight=true), Wendell [P](https://www.safaribooksonline.com/search/?query=author%3A%22Patrick%20Wendell%22&sort=relevance&highlight=true)., Konwinski [A](https://www.safaribooksonline.com/search/?query=author%3A%22Andy%20Konwinski%22&sort=relevance&highlight=true)., Karau [H.](https://www.safaribooksonline.com/search/?query=author%3A%22Holden%20Karau%22&sort=relevance&highlight=true), *Learning Spark,,* [O'Reilly Media, Inc.](https://www.safaribooksonline.com/library/publisher/oreilly-media-inc/), 2015.

**ICT 4035: INFORMATION RETRIEVAL [3 0 0 3]**

**Objectives:**

* To explain basic information storage and retrieval concepts.
* To describe what issues are specific to efficient information retrieval.
* To give applications of alternative search strategies and explain why the particular search strategy is appropriate for the application.

**Abstract:**

Boolean Retrieval Model, Index Construction, Index Compression, Vector Space Model, Evaluation in IR, Relevance Feedback and Query Expansion, Latent Semantic Indexing, Web Search Basics, Web Crawling and Indexes, Link Analysis

**Syllabus:**

**Boolean Retrieval Model**: Building inverted index, Processing Boolean queries, Extended Boolean retrieval model vs ranked retrieval **[2 Hours]**

**Index Construction**: Term vocabulary and posting analysis, Dictionaries & tolerant retrieval, BSBI, SPMI, Distributed indexing, Dynamic indexing **[5 Hours]**

**Index Compression**:

Heap’s and Zipf’s law, Dictionary compression, Postings compression **[4 Hours]**

**Vector Space Model**:

Parametric and zone indexes, Term frequency & weighting, Vector space model for scoring, variants of tf-idf functions  **[4 Hours]**

**Evaluation in IR**:

Computing scores in a complete search system, IR system evaluation, Standard test collections, Evaluation of unranked and ranked retrieval sets, Assessing relevance, Result snippets.**[4 Hours]**

**Relevance Feedback and Query Expansion**:

Relevance feedback and pseudo relevance feedback, Global methods of query reformulation

**[2 Hours]**

**Probabilistic Information Retrieval**:

Probability ranking principle, Binary independence model, Okapi BM25 **[4 Hours]**

**Latent Semantic Indexing**:

Term-document matrices and SVD, Low rank approximation, LSI **[4 Hours]**

**Web Search Basics**:

Advertising as economic model, Search user experience, Index size estimation, Near-duplicates and shingling **[2 Hours]**

**Web Crawling and Indexes**:

Must and Should feature of a web crawler, Crawling, distributing indexes, Connectivity servers

**[3 Hours]**

**Link Analysis**:

Web graph, PageRank, Hubs and Authorities **[2 Hours]**

**Outcomes:**

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

* Read and understand the literature in the area of information retrieval.
* Design and implement a small to medium size information storage and retrieval system.
* Choose right kind of retrieval algorithm for an application/scenario, having considered the strength and weakness of the algorithm

**References:**

1. Manning C. D., Raghavan P., and Schütze H., *Introduction to Information Retrieval*, Cambridge University Press, 2008.
2. Buettcher S., Charles L. A., Clarke, and Cormack G. V., *Information Retrieval: Implementing and Evaluating Search Engines*, MIT Press, 2010.
3. Grossman D. A. and Frieder O., *Information Retrieval: Algorithms and Heuristics,* Springer*,* 2004.

**ICT 4056: MACHINE LEARNING FOR DATA ANALYTICS [3 0 0 3]**

**Course Objectives:**

* To understand the generic principle of learning.
* To introduce the basic principles, techniques, and applications of Machine Learning.
* To apply machine learning concepts for data analytics

**Abstract:**

Introduction to Machine Learning, Mathematical Preliminaries, Supervised Learning-LMS, Linear regression, Logistic regression, GDA, Naive Bayes, SVM, model selection, Learning theory-bias/variance tradeoff, union and Chernoff bounds, VC dimensions, Unsupervised learning-clustering, k-means, Gaussian mixture, factor analysis, PCA, ICA.

**Syllabus:**

**Introduction:**

Basic concepts of machine learning, some day to day examples of machine learning [**1 Hour**]

**Mathematical Preliminaries:**

Review of Linear Algebra, Probability Theory Review, Overview of Convex Optimization, Multivariate Gaussian Distribution, Gaussian Processes **[4 Hours]**

**Supervised Leaning:**

Supervised learning setup, LMS, Linear regression, Logistic regression, Multiple logistic regression model, Interpretation of fitted logistic regression model, Model-building strategies and methods for logistic regression, Assessing the Fit of the model, Perceptron, Exponential family, Generative learning algorithms, Gaussian discriminant analysis, Naive Bayes, Support vector machines, Model selection and feature selection,

Decision trees: Non-linearity, Region selection, Defining loss function, Factors to consider: Categorical variable, Regularization, Runtime, Lack of additive structure, ID3 algorithm,

Ensemble methods: Bagging, Boosting. Evaluating and debugging learning algorithms  **[20 Hours]**

**Learning Theory:**

Bias/variance tradeoff, Union and Chernoff and Hoeffding bounds, VC dimension, Practical advice on how to use learning algorithms [3 **Hours**]

**Unsupervised Learning:**

Clustering, K-means, EM, Mixture of Gaussians, Factor analysis, PCA (Principal Component Analysis), ICA (Independent Component Analysis) [**8 Hours**]

**Course Outcomes:**

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

* Understand machine learning algorithms and their use in data-driven knowledge discovery.
* Identify the suitability of discriminative or generative supervised model for a given problem
* Choose an appropriate supervised or unsupervised model for a given data analytics-based learning problem
* Apply diagnostics for debugging learning algorithms.

**References:**

1. Kevin P Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.
2. Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar., *Foundations of Machine Learning*, MIT Press, 2012.
3. David W.Hosmer, Stanley Lemeshow, and Ridney X.Sturdivant, *Applied Logistic Regression*, 3ed, Wiley, 2013.
4. Christopher M.Bishop., *Pattern Recognition and Machine Learning (2e),* Springer, 2013.

**ICT 4036: SEMANTIC WEB [3 0 0 3]**

**Objectives:**

* To provide introduction to the Semantic Web technologies and their applications.
* To provide introduction to the fundamental concepts of semantic web: ontology languages, design, reasoning and querying.
* To build and analyse ontologies using an ontology editor.

**Abstract:**

Semantic Web Vision, Layered approach, Describing Web Resource : RDF data models, syntax, semantics, schema, RDFS, Direct inference system, RDF Data structures, Containers and colletcions, Querying Semantic Web: SPARQL matching patterns, filters, querying schemas Ontology and Information Systems L use of ontologies, types, design principles, methodologies, Ontology Languanges : OWL2, OWL2 profiles, Ontology Reasoning: Monotonic rules, Rule interchange format, Semantic web rules languages, RuleML, Ontology Design and Management: Types, purposes, creating ontology manually, reusing existing, mapping, Ontology Programming : Programming in Go

**Syllabus:**

**Semantic Web Vision**: Introduction, semantic web technologies, a layered approach, **[2 Hours]**

Reference : Reference number 1

**Describing Web Resource**:

RDF-data model, syntaxes, RDFS-adding semantics, RDF schema, RDF and RDF schema in RDFS, Axiomatic schematics for RDF and RDF schema, Direct inference system for RDF and RDFS, RDF Data Structures, RDF Containers, RDF Collections, Example on containers and collections **[8 Hours]**

Reference : Reference number 1, 3, 5

**Querying Semantic Web**:

SPARQL infrastructure, matching patterns, Filters, Constructs for dealing with open world, Organizing result sets, other forms of SPARQL queries, Querying schemas, Examples on SPARQL **[6 Hours]**

Reference : Reference number 1, 2

**Ontology and Information Systems**:

Ontology, Uses of Ontologies, Types, Architectures, Design Principles, Development Approaches, Design Methodologies  **[2 hours]**

Reference : Reference number 1

**Ontology Languages**:

OWL2, Requirement of ontology language, Compatibility of OWL2 with RDF/RDFS, OWL language, OWL2 profiles **[6 Hours]**

Reference : Reference number 1

**Ontology Reasoning**:

Monotonic rules, OWL2 RL, Rule interchange format (RIF), Semantic web rules language (SWRL), Rules in SPARQL, Non monotonic rules, Rule markup language (RuleML) **[4 Hours]**

Reference : Reference number 1

**Ontology Design and Management**:

Types, Purpose, Constructing ontologies manually, Reusing existing ontologies, Semi-automatic ontology acquisition, Ontology mapping, Exposing relational databases, Semantic web application architecture, Creating own ontology **[6 Hours]**

Reference : Reference number 6, 1

**Ontology Programming**:

Pure Functional Programming in Go, Manipulating collections **[2Hours]**

Reference : Reference number 4

**Outcomes:**

* By the end of this course, the student should be able to
* Understand the conceptual structure of the semantic web technology
* Understand the concepts of metadata, semantics of knowledge, ontology, XML-based syntax, web ontology language (OWL).
* Describe logic semantics, inference with OWL
* Use ontology engineering approaches in semantic applications
* Comprehend the content of research paper on semantic web

**References:**

1. Grigoris Antoniou, Paul Groth, Frank van vanHarmelen and Rinke Hoekstra, A Semantic Web Primer, MIT Press, 2012.
2. Bob DuCharme, *Learning SPARQL*, 2nd edition, O’REILLY, 2013.
3. Brain McBride, Dan Brickley, R.V. Guha, *‘RDF Schema 1.1*’, 2014, [Online], Available: <https://www.w3.org/TR/rdf-schema>, [Accessed: 29-Jan-2018]
4. Lex Sheehan, Learning Functional Programming in Go, Packt Publishing, 2017, ISBN : 978-1-78728-139-4
5. Frank Manola, Eric Miller, David Veckett, Ivan Herman, 'RDF Primer Turtle Version', 2007. [Online]. Available: <https://www.w3.org/2007/02/turtle/primer/> . [Accessed: 29- Jan- 2018]
6. Vandana Kabilan Ontology for Information Systems (O4IS) Design Methodology(Doctoral dissertation, 2007.

**IV.SOFTWARE SYSTEM DESIGN**

**ICT 4037: ADVANCED SOFTWARE ENGINEERING [3 0 0 3]**

**Objectives:**

* To discuss recent advances in software engineering practices and process models
* To understand the web engineering activities to develop web app
* To learn software engineering concepts and their applicability in practice

**Abstract:**

Specialized process models-Component based development, Formal methods model, Aspect Oriented S/W Development; Agile view of process-Agile process, agile process models; Applying Web Engineering- initiating web app project, analysis for web apps, design for web apps, testing for web apps; Formal methods-Concepts; Formal specifications- Specification Qualities, Classification of Specification Styles, Descriptive Specifications- Logic and Algebraic Specifications, Operational Specifications-DFD, FSM, Petri Nets, Introduction to *Z,* Cleanroom Software Engineering, Software Engineering: Security Engineering.

**Syllabus:**

**Specialized process models-** Component based development-The CBSE process, Domain engineering; Aspect Oriented S/W Development; Agile view of process-Agile process, agile process models. **[6 hours]**

**Formal methods-** Basics, Mathematical preliminaries, applying mathematical notations for specifications, Formal specification language-OCL/Z, Operational specifications- FSM, Petri Nets **[8 hours]**

**Applying Web Engineering-** Web engineering- analysis for web apps, design for web apps, testing for web apps-strategy and techniques.  **[10 hours]**

**Cleanroom s/w engineering-** The cleanroom approach, Functional specification, cleanroom design, cleanroom testing. **[4 hours]**

**Reengineering-** Business process reengineering, Software reengineering, Reverse engineering, Restructuring, Forward engineering. **[5 hours]**

**Security Engineering-** Analyzing security requirements, Security and Privacy in an online world, Security engineering analysis, Security Assurance. **[3 hours]**

**Course Outcomes:**

**Upon completion of the course, the students will be able to**

* Understand the specialized process models
* Understand the formal methods, their applicability in practice
* Apply web engineering best practices to develop web app.
* Appraise advanced software engineering concepts and their applicability in practice
* Apply the most relevant methods, best practices, tools, frameworks/processes, technologies for development of a software product

**References:**

1. Pressman R. S., *Software Engineering-A practitioner’s approach* *(8e)*, McGraw-Hill Publications, 2014.
2. Wolfgang R., *Understanding Petri Nets Modelling Techniques, Analysis, Methods, Case Studies*, Springer-Verlag, 2013.
3. Ghezzi, Jazayeri M., and Mandrioli D., *Fundamentals of Software Engineering* *(2e)*, Pearson India, 2015 (Paper Back).

**ICT 4038: SOFTWARE ARCHITECTURE [3 0 0 3]**

**Objectives:**

* To understand the concepts, principles, methods in software architecture.
* Learn Architectural patterns and Reference models
* Discuss Quality Attributes, relationship of tactics to architectural patterns
* Learn documenting software architectures
* Discuss optimal architectural choices for implementing a software product, regardless of its complexity and scale.

**Abstract:**

The Architecture Business Cycle, Software Processes and the Architecture Business Cycle, Architectural Patterns, Reference Models, and Reference Architectures, Architectural Structures and Views, Understanding Quality Attributes, Functionality and Architecture, Architecture and Quality Attributes, Achieving Qualities, Tactics-Availability, Modifiability , Performance , Security, Testability, Usability, Relationship of Tactics to Architectural Patterns, Designing the Architecture, Documenting Software Architectures, Reconstructing Software Architectures, Information Extraction, Database Construction, View Fusion, Reconstruction, The 4+1 Views, General UML features, Transaction and Data Design, Architectural patterns :Interactive systems , Adaptable systems, Design Patterns.

**Syllabus:**

**Introduction of Software Architecture-** The Architecture Business Cycle, Software Processes and the Architecture Business Cycle **[3 Hours]**

**Architectural Patterns-** Architectural Patterns, Reference Models, and Reference Architectures, Architectural Structures and Views  **[4 Hours]**

**Quality attributes-** Understanding Quality Attributes, Functionality and Architecture, Architecture and Quality Attributes, Achieving Qualities, Case Study **[4 Hours]**

**Quality Tactics-** Introducing Tactics, Availability Tactics, Modifiability Tactics, Performance Tactics, Security Tactics, Testability Tactics, Usability Tactics, Relationship of Tactics to Architectural Patterns **[6 Hours]**

**Architecture Documentation and Reconstruction-** Documenting Software Architectures, Uses of Architectural Documentation, Reconstructing Software Architectures, Information Extraction, Database Construction, View Fusion, Reconstruction. **[3 Hours]**

**Architectural Details -** Roles of the Software Architect, Relationship to other key roles in development organization, Architectural viewpoint, Skills and Background for the Architect, Software Architecture and the Development Process, Architecture and agile processes, Requirements Management, Effective Technical Meetings, Traps and Pitfalls of the Software Architecture Process Activities, Example System Overview  **[2 Hours]**

**UML-** The 4+1 Views, General UML features, Behavioral View, Component instance diagrams, Class and subsystem diagrams, Deployment diagrams, State-chart diagrams **[5 Hours]**

**Transaction and Data Design-** Transaction and Data Design, Data Model Design. [**3 Hours]**

**Architectural and Design Patterns-** Architectural patterns: Interactive systems, Adaptable systems, Design Patterns .  **[6 Hours]**

**Course Outcomes:**

**Upon completion of the course, the students will be able to,**

* Recognize the major architectural styles, design patterns
* Identify the quality attributes of a system at architectural level
* Use well-understood paradigms for designing new systems
* Document a software architecture using relevant approach and descriptions

**References:**

1. Len B., Clements P., and Kazman R., *Software Architecture in Practice* *(2e)*, SEI Series in Software Engineering, 2012.
2. Buschmann F., Meunier R., Rohnert H., Sommerlad P., and Stal M., *Pattern-Oriented Software Architecture, A System of Patterns* *(4e)*, John Wiley and Sons, 2008.
3. Cervantes H., Kazman R., *Designing Software Architectures: A Practical Approach* *(1e)*, Addison-Wesley Professional, 2016.

**ICT 4039: SOFTWARE PROJECT AND QUALITY MANAGEMENT [3 0 0 3]**

**Objectives:**

* To learn Software Project Models and Software Management Concepts
* Discuss various methods of Cost Estimation.
* Introduce students to software management and metrics.
* To learn software quality management and models

**Abstract:**

Introduction to Project Management- Project Management activities, Project Estimation and Planning-Software effort estimation techniques, Activity Planning, Network planning models Project Risk Management and Monitoring- Resource Scheduling, Cost Monitoring, Software Quality Management- Software Quality Attributes and Specification, Defect Prevention, Reduction, and Containment, Quality Management and Assurance- Total Quality Management, Software Verification, Validation & Testing, Quality Standards-ISO, Six Sigma concepts.

**Syllabus:**

**Introduction to Project Management**

Project, Types of Project, Project Management activities, problems with the software projects, management control, Stakeholders, Requirement specifications, Information and control in organizations, Step wise project planning, Project Evaluation- Strategic assessment and technical assessment, Selection of an appropriate project approach **[04 hours]**

**Project Estimation and Planning**

Introduction of estimation, Problems with over and under estimates, Basis of software estimating, Software effort estimation techniques, Expert judgement, Function point analysis, Object point analysis, A procedural code oriented approach, COCOMO: a parametric model, Activity Planning- Objectives, Project Schedules, Projects and activities, Sequencing and scheduling activities, Network planning models, Formulating a network model and identifying the critical path, Shortening the project duration **[08 hours]**

**Project Risk Management and Monitoring**

Risk Management- Project Risks, Managing Risks, Identification and Analysis Risks, Evaluating Risks to the Schedule, Resource Allocation- Nature of Risk, Identifying resource requirements, Scheduling resources, Creating critical path, Monitoring and Control- Creating the framework, Collecting the data, Visualizing the progress, Cost Monitoring, Earned Value, Prioritizing monitoring, Change Control  **[05 hours]**

**Software Quality Management**

Defining Software Quality, Software Quality Attributes and Specification, Cost of Quality, Defects, Faults, Failures, Defect Rate and Reliability, Defect Prevention, Reduction, and Containment, Overview of Different Types of Software Review, Introduction to Measurement and Inspection Process, Documents and Metrics. Software Quality Factors – Software Quality Components – Software Quality Plan – Software Quality Metrics – Software Quality Costs – Software Quality Assurance Standard – Certification – Assessment.  **[08 Hours]**

**Quality Management and Assurance**

Software Quality Management and Models, Software Quality Assurance -Quality Planning and Control, Quality Improvement Process, Evolution of Software Quality Assurance (SQA), Major SQA Activities, Major SQA Issues, Zero Defect Software, SQA Techniques, Statistical Quality Assurance, Total Quality Management, Quality Standards and Processes. Software Verification, Validation & Testing **[8 Hours]**

**Quality Standards**

Introduction to ISO 12207, Approach to software life cycle data and life cycle processes, Acquisition process, Supply process, Development process, CMMI - Six Sigma concepts. **[3 Hours]**

**Course Outcomes:**

**Upon completion of the course, the students will be able to,**

* Understand the project management and software quality management aspects
* Appraise the SQM, models and metrics for software quality assurance.
* Appraise the project activities and its management for software development.
* Perform cost estimation for software product development.

**References:**

1. Hughes B., Cotterell M., Mall R., *Software Project Management* *(6e)*, McGraw Hill, 2017.
2. Tian J., *Software Quality Engineering (SQE)*, Wiley, 2006.
3. Jalote P., *Software Project Management in Practice* *(1e)*, Addison Wesley Professional, edition, 2009.
4. Pressman R. S., *Software Engineering: A practitioner’s approach* *(8e)*, McGraw Hill, 2014.
5. Kelkar S. A., *Software Project Management: a concise study* *(3e)*, PHI Learning-New Delhi, 2013.
6. Kan S. H., *Metrics and Models in Software Quality Engineering* *(2e)*, Pearson, 2008.

**ICT 4040: SOFTWARE CONSTRUCTION [3 0 0 3]**

**Objectives:**

* To gain knowledge about software construction process
* To acquire skills to design a high quality code
* Learn to implement a well-organized high quality code
* To learn about different methods for code improvements.
* To learn effective documentation of the code.

**Abstract:**

Introduction to software construction; Creating High Quality Code-Design in construction, Working classes, High Quality Routines, Defensive Programming, The Pseudo code Programming Process; Variables – General issues in Using Variables, The Power Of Variable Names, Data Types; Statements- Organizing Straight Line Code, Using conditionals, Controlling loops, Unusual Control Structures, Table Driven Methods, General Control Issues; Code Improvements- Software Quality Landscape, Collaborative Construction, Developer Testing, Debugging, Refactoring, Code Tuning Strategies, Code Tuning Techniques; Software Craftsmanship-Layouts and Style, Self-Documenting Code, Personal Character, Themes in Software Craftsmanship.

**Syllabus:**

**Laying the foundation:**

Introduction to software construction, Metaphors for a Richer Understanding of software development, Measure Twice Cut Once: Upstream Prerequisites, Key Construction Decisions **[3 Hours]**

**Creating High Quality Code:**

Design in construction, Working classes, High Quality Routines, Defensive Programming, The Pseudocode Programming Process. **[5 Hours]**

**Variables:**

General issues in Using Variables, The Power Of Variable Names, Fundamental Data Types, Unusual Data Types **[4 Hours]**

**Statements:**

Organizing Straight Line Code, Using conditionals, Controlling loops, Unusual Control Structures, Table Driven Methods, General Control Issues **[7 Hours]**

**Code Improvements:**

Software Quality Landscape, Collaborative Construction, Developer Testing, Debugging, Refactoring, Code Tuning Strategies, Code Tuning Techniques **[9 Hours]**

**System considerations:**

How Program Size Affects Construction, Managing Construction, Integration, Programming Tools **[5 Hours]**

**Software Craftsmanship:**

Layouts and Style, Self-Documenting Code, Personal Character, Themes in Software Craftsmanship. **[3 Hours]**

**Outcomes :**

**Upon completion of the course, the students will be able to**

* Acquire a better understanding of the process involved in software construction.
* Design a high quality code
* Write a high quality code
* Get an insight about different programming tools.
* Provide a proper layout to the code along with appropriate documentation.

**References:**

1. McConnell S., Code Complete: A practical Handbook for Software Construction (2e), Microsoft Press, 2004.
2. Meyer B., Object Oriented Software Construction (2e), Prentice Hall, 1997.
3. Martin R. C., Clean Code: A handbook of Agile Software Craftsmanship, Prentice Hall, Pearson Education, 2012.
4. Martin R. C., Clean Architecture: A Craftsman's Guide to Software Structure and Design (1e), Prentice Hall, 2017.

**V. MATERIAL SCIENCE :**

PHY 4051 : Physics of Law Dimensional Materials

PHY 4052: Physics of Photonic & Energy storage devices

CHM 4051: Chemical Bonding

CHM 4052: Chemistry of Carbon compound

**VI. BUSINESS MANAGEMENT**

**HUM 4051: FINANCIAL MANAGEMENT [2 1 0 3]**

**Course Outcomes:**

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

* Calculate the value of various financial assets such as annuities, bonds, and stocks.
* List the primary sources of capital and incorporate their cost when making investment decisions such as debt, preferred stock, and common stock.
* Estimate project cash flows by using the concepts of time value of money.
* Estimate cash flows from a project, including operating, net working capital, and capital spending.
* Understand the basic accounting principles.
* Apply the knowledge of book keeping, costing and accounting in day-to-day activities.
* Prepare budgets and costing to industrial scenarios.

**Abstract:**

Introduction and objectives of financial management, Evolution of corporate finance and its responsibilities. Types of accounts, Golden rules of accounting, Preparation of journal, Ledger, Trial balance and final accounts. Sources of long term finance, Characteristics of equity capital, Preference capital, Debenture capital & Term loans. Valuation of securities, Concepts, Bond valuation and related models, Bond value theorems, Yield to maturity. Equity valuation; Dividend capitalization approach, Leverage, Operating leverage, Financial leverage, Total leverage, Indifference point analysis. Working capital management, Capital budgeting: appraisal criteria, pay-back period, Average rate of return, Net present value, Benefit cost ratio and internal rate of return. Risk analysis in capital budgeting, Cost of capital: introduction, cost of debt capital, Preference capital and Equity capital, Weighted average cost of capital, Determination of proportions, Cash management, and Dividend decisions.

**Introduction to Financial Management**

Introduction and objectives of financial management, Functions of financial management, Evolution of corporate finance and its responsibilities. **[2 Hours]**

**Principle of Accountancy**

Accounting concepts, Single and double entry book keeping, Types of accounts, Golden rules of accounting, Preparation of journal, ledger, trial balance, and final accounts. **[5 Hours]**

**Sources of Long Term Finance**

Characteristics of equity capital, Preference capital, Debenture capital, and Term loans**.[3 Hours]**

**Valuation of Securities**

Concepts, Bond valuation and related models, Bond value theorems, Yield to maturity, Equity valuation, Dividend capitalization approach, and Ratio approach. **[6 Hours]**

**Leverage**

Operating leverage, financial leverage, Total leverage, and Indifference point analysis. **[2 Hours]**

**Working Capital Management**

Concepts of working capital, Objectives and need for working capital, Operating cycle, Approaches for working capital management, and Estimation of working capital. **[4 Hours]**

**Capital budgeting**

Appraisal criteria, Pay-back period, Average rate of return, Net present value, Benefit cost ratio and Internal rate of return, and Risk analysis in capital budgeting. **[5 Hours]**

**Cost of Capital**

Introduction, Cost of debt capital, preference capital and equity capital, Weighted average cost of capital, and Determination of proportions. **[3 Hours]**

**Cash Management**

Meaning and importance of cash management, Motives of holding cash, Objective of cash management, and Models for determining optimal cash needs. **[3 Hours**]

**Dividend Decisions**

Introduction, Traditional approach, Dividend relevance model, Stability of dividends, Forms of dividends, and Stock split.  **[3 Hours]**

**References:**

1. Prasanna Chandra, Fundamentals of Financial Management, Tata McGraw Hill, Delhi, 2006.
2. I M Pandey, Financial Management, Vikas Publishing house, Delhi, 2007.
3. Subir Kumar Banerjee, Financial Management, Sultan Chand & Co., Delhi.1999.
4. ICFAI, Corporate Financial Management, ICFAI, Hyderabad, 2003.
5. Maheshwari S.N, Financial Management, Sultan Chand & Co., Delhi, 2002

**HUM 4052: HUMAN RESOURCE MANAGEMENT [2 1 0 3]**

**Course Outcomes:**

On completion of this course the student should be able to:

* Understand Human Resource Management, its development, purpose and organization
* Understand the operational and strategic purpose of HRM
* Comprehend the importance of interdepartmental relationships
* Evaluate employee performance
* Analyse, test, and evaluate grievances and conflicts
* Develop team spirit, ethical consciousness, and workplace skills
* Demonstrate the ability to incorporate and use modern technology in collecting and analysing data from a variety of sources
* Develop an awareness of our interdependence as global citizens combined with an understanding of the history, culture, diversity, and commonality of life’s experiences.

**Abstract:**

Evolution and development: Introduction, Scope of HRM, Objectives of HRM, Functions of HRM, Activities of HRM, Managerial skills and roles, HRD organization and responsibilities; Evolution of HRM, Influence of various factors on HRM, Theories of HRM; Human resource planning- Introduction, Strategic considerations, Nature and scope, Human resources inventory, Forecast, Job analysis, Job description, Job specification, Job evaluation, Employment stability; Recruitment, Selection, Placement and induction, Scientific selection, Policy, Process, Tests, Interview, Work history, References, Provisional selection, Medical/Physical examinations, Final selection, Employment; Induction, & socialization - Placement policy, Induction programs, socialization programmes; Training and development - Basic concepts, Employees training, Training process, Planning, Preparation of trainees, Implementation, Performance evaluation, Follow-up training; Management executive development and Career development - Basic concepts, Stages of career development, Career development programmes; Promotion transfers and separations; Wages and salaries administration; Discipline and grievances, Industrial and labour relations and Trade unionism, Collective bargaining, Industrial health, Performance appraisal and Merit rating.

**The Strategic Role of HRM:** Nature, scope, objectives, importance and functions of HRM; Human resource as an asset in organization, Evolution of the concept of HRM, Human resource management in India; Strategic HRM, Strategic roles of HR manager, and Qualities of HR Manager. **[6 Hours]**

**Job Analysis & Design:** Job Analysis – Meaning, uses, process and methods of collecting data for job analysis, Job description, Job specifications, Factors affecting job design, Techniques of job design. **[6 Hours]**

**Job Evaluation and Human Resources Planning:** Objectives of job evaluation; Job evaluation methods; Advantages and limitations of job evaluation; Human resources planning, Need for human resources planning; Process of human resources planning; Human resource planning system; Responsibility for human resource planning, and Employment stability. **[6 Hours]**

**Employment, Induction, & Socialization**: Placement policy, Induction programs, and socialization programmes. **[4 Hours]**

**Training and Development:** Basic concepts; Employees training- Training process, Planning, Preparation of trainees, Implementation, Performance evaluation, and Follow-up training; The steps in Training Process; Career and succession planning- Career stages, Career development, Career management succession planning; Case discussion on succession planning, Performance appraisal, and Merit rating.  **[10 Hours]**

**Labor Relations and Employee Security:** Promotion transfers and separations, Wages and salaries administration, Discipline and grievances, Industrial and labor relations, Trade unionism, Collective bargaining, and Industrial health. **[4 Hours]**

**Reference Books:**

1. T.V. Rao and Pereira D, Recent experiences in Human Resources Development, Oxford and IBH Publishing, 1986.
2. Subbrao A., Essentials of Human Resource Management and industrial Relations, Himalaya Publishing House, 1999.
3. N G Nair and Latha Nair, Personnel Management and Industrial Relations, S. Chand Company, 1995.
4. Virmani B R; Rao Kala, Economic restructuring technology transfer and human resource development, Response books, 1997.
5. Pareek Udai et al., Human Resource Development in Asia: Trends and Challenges, Oxford and IBH Publishing, 2002.
6. Michael Armstrong., A handbook of Human Resource Management Practice, (10e), Kogan Page limited.
7. Gary Dessler & Biju Varkkey, Human Resource Management, Pearson education, 2011.

**HUM 4053: MARKETING MANAGEMENT [2 1 0 3]**

**Course Outcomes:**

On completion of this course the student should be able to:

* Understand key marketing concepts, theories and the importance of customer satisfaction and retention
* Analyse the factors influencing buying behaviour, competitive environment and strategies
* Identify and understand market segments and positioning, product life cycle, and new product development
* Understand the mechanisms of pricing, marketing channels, and design
* Integrate marketing communication for a product or service.

**Abstract:**

Defining marketing for the Twenty-first Century, Scope of marketing, Marketing concepts; Adapting marketing to the New Economy: Major drivers of the new economy, how business practices are changing. Building Customer Satisfaction, Value, and Retention: Defining customer value and satisfaction, Corporate and division strategic planning, Business unit strategic planning. Market Demand: Components of a modern marketing information system. Scanning the Marketing Environment: Analysing needs and trends in the microenvironment. Consumer Markets: Factors influencing buying behaviours. Business Markets: The business market versus the consumer market, major influences on buying decisions, institutional and government markets. Dealing with the Competition: Identifying competitors, analysing competitors. Market Segments: Patterns of market segmentation, Segmenting consumer and business markets. Product Life Cycle: Product life-cycle marketing strategies. New Market Offerings: Challenges in new-product development, Organizing new-product development, New-product development process, Consumer-adoption process. Designing and Managing Services: Characteristics of services, Managing product support services. Price Strategies: Setting the price, Adapting the price, Initiating and responding to price changes. Retailing, Wholesaling: Trends in retailing, Wholesaling, Wholesaler marketing decisions. Integrated Marketing Communications: The major modes of communication, and Marketing communications mix.

**Introduction:** Defining marketing for the twenty-first century, Scope of marketing, Marketing concepts and tools, Company orientations toward the marketplace, and Societal marketing concept. **[02 Hours]**

**Adapting Marketing to the New Economy:** Major drivers of the new economy, changes in business practices. **[02 Hours]**

**Building Customer Satisfaction, Value, and Retention:** Defining customer value and satisfaction, Nature of high-performance businesses, Attracting and retaining customers, Customer profitability, and Company profitability. **[03 Hours]**

**Winning Markets Through Market-Oriented Strategic Planning:** Corporate and division strategic planning and Business unit strategic planning. **[02 Hours]**

**Gathering Information and Measuring Market Demand:** Components of a modern marketing information system and Database marketing. **[02 Hours]**

**Scanning the Marketing Environment:** Analysing needs and trends in the microenvironment and Identifying and responding to the major micro environmental forces. **[02 Hours]**

**Analysing Consumer Markets and Buying Behaviour:** Factors influencing buying behaviours and Buying decision process. **[02 Hours]**

**Analysing Business Markets and Business Buying Behaviour:** Organizational buying, Business market versus the consumer market, Buying situations, Major influences on buying decisions, Institutional and government markets. **[03 Hours]**

**Dealing with the Competition:** Identifying competitors, analysing competitors, designing competitive strategies, balancing customer, and competitor orientation. **[03 Hours]**

**Identifying Market Segments and Selecting Target Markets:** Levels and patterns of market segmentation, Patterns of market segmentation, Market segmentation procedure, and segmenting consumer and business markets. **[03 Hours]**

**Positioning and Differentiating the Market offering through the Product Life Cycle:** Competitive differentiation tools, Developing and communicating a positioning strategy, Product life-cycle marketing strategies, and Stages in market evolution. **[03 Hours]**

**Developing New Market Offerings:** Categories of new products, Challenges in new-product development, Organizing new-product development, Managing the new-product development process, The consumer-adoption process; Designing and managing services categories of service mix, Characteristics of services, and Managing product support services. **[03 Hours]**

**Developing Price Strategies and Programs:** Setting the price, Adapting the price, and Initiating and responding to price changes. **[02 Hours]**

**Managing Retailing, Wholesaling, and Market Logistics:** Retailing and types of retailers, Trends in retailing, Wholesaling, Wholesaler marketing decisions. **[02 Hours]**

**Designing and Managing Integrated Marketing Communications:** The five major modes of communication; Developing effective communications, and Deciding on the marketing communications mix. **[02 Hours]**

**Reference Books**:

1. Philip Kotler, Marketing Management – Analysis, Planning, Implementation and Control, Prentice Hall of India Private Limited, New Delhi, 2000.
2. ICFAI, Marketing Management, ICFAI, Hyderabad, 2003.
3. Varshney R L and Gupta S L, Marketing Management, Sultan Chand & Sons, New Delhi, 2004.
4. Adrian Palmer, Principles of Marketing, Oxford University Press, New York, 2000.

**HUM 4054: OPERATIONS AND SYSTEMS MANAGEMENT [2 1 0 3]**

**Course Outcomes:**

On completion of this course the student should be able to:

* Understand and analyse the importance of operations for a firm.
* Demonstrate decision making ability in core areas of operations such as demand forecasting and capacity planning.
* Develop operations plans to execute business operations on a short run.
* Understand the basics concepts of supply chain management.
* Comprehend the importance of quality and lean principles in operations.

**Abstract:**

Introductions to operations management – process view and supply chain view, Types of production activities, Competitive priorities and capabilities. Break-even analysis, Evaluating services or products, Evaluating processes - make or buy decision, Decision making under risk, and decision trees. Introduction to forecasting, Importance and uses of forecasting, Demand patterns, Demand management options, Judgement methods, Causal methods - linear regression, time series method – Naïve method, Moving average, Weightage moving average, and Exponential smoothing curve. Planning long-term capacity, Measures of capacity and utilization, Economies of scale, Diseconomies of scale, Capacity timing and Sizing strategies, Sizing capacity cushions, Timing and sizing expansion – Expansionist strategy, Wait and see strategy, and a Systematic approach to long term capacity decision. Levels inoperations planning and scheduling across the organization, Sales and operation planning strategies- Chase strategy, Level strategy, Operations planning using linear programming technique, scheduling job and facility scheduling, and work for scheduling. Theory of constraints, Managing bottle necks in manufacturing and service processes, Identifying bottle necks, Relieving bottle necks, Drum buffer rope system, and Managing constraints in a line system. Supply chain design across the organization, Supply chains for services and manufacturing, Measures of supply chain performance - Inventory measures, financial measures, Inventory and supply chains - pressures for small inventories, Pressures for large inventories, Types of inventory, Inventory reduction tactics, and Inventory placement. Costs of quality, Total quality management, Acceptance sampling, Statistical process control - Control charts, and Process capability. Continuous improvement using lean systems, Different types of wastes, Strategic characteristics of a lean system, Designing lean system layout, and Kanban system.

**Competing with operations:** Introductions to operations management – Process view and supply chain view, Types of production activities, Competitive priorities and capabilities. **[02 Hours]**

**Decision theory** - Break-even analysis, Evaluating services or products, Evaluating processes - Make or buy decision, Decision making under risk, and Decision trees. **[3 Hours]**

**Forecasting:** Introduction to forecasting, Importance and uses of forecasting, Demand patterns, Demand management options, Judgement methods, Causal methods - Linear regression, Time series method – Naïve method, Moving average, Weightage moving average, and Exponential smoothing curve. **[4 Hours]**

**Capacity planning:** Planning long-term capacity, Measures of capacity and utilization, Economies of scale, Diseconomies of scale, Capacity timing and sizing strategies, Sizing capacity cushions, Timing and sizing expansion – Expansionist strategy, Wait and see strategy, and Systematic approach to long term capacity decision. [**5 Hours]**

**Planning and scheduling operation:** Levels inoperations planning and scheduling across the organization, Sales and operation planning strategies- Chase strategy, Level strategy, Operations planning using linear programming technique, Scheduling job and facility scheduling, and Work for scheduling. [**5 Hours]**

**Constraints management:** Theory of constraints, Managing bottle necks in manufacturing and service processes, Identifying bottle necks, Relieving bottle necks, Drum buffer rope system, and managing constraints in a line system. [**5 Hours]**

**Supply Chain Management/Design:** Supply chain design across the organization, Supply chains for services and manufacturing, measures of supply chain performance - inventory measures, financial measures, inventory and supply chains - pressures for small inventories, pressures for large inventories, types of inventory, inventory reduction tactics, and inventory placement.

[**5 Hours]**

**Quality and Performance:** Costs of quality, Total quality management, Acceptance sampling, statistical process control - Control charts, and Process capability. [**3 Hours]**

**Lean Systems:** Continuous improvement using lean systems, Different types of wastes, Strategic characteristics of a lean system, Designing lean system layout, and Kanban system. [**4 Hours]**

**Reference Books:**

1. Krajewski L. J., Ritzman L. P., Malhotra M., and Srivastava S. K., Operations Management, (11e), Pearson Education (Singapore) Pvt. Ltd., Delhi, 2016..
2. Heizer J. and Render B., Operations Management, (11e), Pearson Education India, 2016.
3. Khanna R. B., Production and Operations Management,, (2e), PHI Learning Private Limited, 2015.

**VII. Computational Mathematics**

**MAT 4051: Applied Statistics and Time Series Analysis**

**MAT 4052: Computational Linear Algebra**

**MAT 4053: Computational Probability and Design of Experiments**

**MAT 4054: Graphs and Matrices**

**Other Electives**

## **ICT 4045: CLOUD COMPUTING [3 0 0 3]**

**Objectives:**

* To learn the fundamental concepts and various delivery, deployment models in cloud computing.
* To understand the demand and need based allocation of resources to the applications running in cloud
* To familiarize with the role of virtualization layer, various types of virtualization techniques.
* To understand various techniques to secure the cloud resources from unauthorized access.

**Abstract:**

Introduction, Cloud infrastructure, Cloud computing delivery models and services, Cloud computing at Amazon, The Google perspective, Microsoft Windows Azure, Application paradigms, Architectural styles of cloud computing, Cloud resource management and scheduling, Cloud resource virtualization, Types of virtualization, Understanding hypervisors, Virtual machine and its components, Resource management, Memory ballooning, Thin virtual provisioning, Storage tiering, Virtual LAN, VLAN trunking, VLAN tagging, Business continuity and cloud management, Virtual machine fault tolerance, Virtual machine replication methods, Cloud security, Virtual machine security, Access control and identity management, Cloud tools: Eucalyptus, OpenNebula/OpenStack, CloudSim.

**Syllabus:**

**Introduction, Cloud Infrastructure:**

Cloud computing, Cloud computing delivery models and services, Ethical issues, Cloud vulnerabilities, Major challenges, Cloud computing at Amazon, The Google perspective, Microsoft Windows Azure, Open-source software platforms for private clouds, Cloud storage diversity and vendor lock-in, Energy use and ecological impact, Service level agreements, User experience and software licensing. **[4 Hours]**

**Application Paradigms:**

Challenges of cloud computing, Architectural styles of cloud computing, Workflows: Coordination of multiple activities, Coordination based on a state machine model: The Zookeeper, The Map Reduce programming model. **[4 Hours]**

**Cloud Resource Management and Scheduling:**

Policies and mechanisms for resource management, Feedback control based on dynamic thresholds, Coordination of specialized autonomic performance managers, Resourcing bundling: Combinatorial auctions for cloud resources, Scheduling algorithms for computing clouds, Fair queuing, Start-time fair queuing, Borrowed virtual time, Cloud scheduling subject to deadlines, Scheduling MapReduce applications subject to deadlines, Resource management and dynamic scaling. **[5 Hours]**

**Cloud Resource Virtualization:**

Virtualization, Layering and virtualization, Virtualization Overview, Virtualized Data Center (VDC) – Compute, Types of Virtualization, Understanding Hypervisors, Virtual Machine and its Components, Resource Management, Share, Limit and Reservation, Optimizing Memory Resource, Memory Ballooning, Virtual Machine Affinity, Physical to Virtual Conversion: Hot and Cold Conversion Process, Virtualized Data Center (VDC) – Storage, Storage Virtualization at different Layers, Virtual Machine Storage Options and Considerations, Virtual Provisioning, Storage Tiering, Virtualized Data Center (VDC) – Networking, Components of VDC network infrastructure, Virtual Network Components, Virtual LAN, VLAN Trunking, VLAN Tagging, Network Traffic Management, Virtualized Data Center (VDC) - Desktop and Application, Performance comparison of virtual machines, The dark side of virtualization, Case Study: Xen, a VMM based par virtualization. **[9 Hours]**

**Business Continuity and cloud management:**

Overview, virtual machine fault tolerance, NIC teaming, backup optimization, Virtual machine replication methods, Service failure, Virtual infrastructure management software, Cloud service management. **[5 Hours]**

**Cloud Security:**

Information Security, Basic Terminology, Cloud security risks, Security concerns and Threats, Privacy and privacy impact assessment, Trust, Operating system security, Virtual machine Security, Security of virtualization, Security risks posed by shared images, Security risks posed by a management OS, A trusted virtual machine monitor,Access Control and Identity Management in Cloud, Governance, Risk and Compliance, Virtualization Security Management, Trusted Cloud Computing **[6 Hours]**

**Cloud Tools:**

Overview of cloud software: Eucalyptus, Open Nebula/Open stack, CloudSim Framework: Modelling and simulating the cloud environment **[3 Hours]**

**Outcome:**

**Upon completion of the course, the students will be able to:**

* Explain the fundamental concepts in cloud computing.
* Describe various resource management techniques.
* Explain various virtualization techniques.
* Compare various techniques used to secure cloud resources.

References:

1. Dan C Marinescu, *Cloud Computing Theory and Practice*, Elsevier 2013
2. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, *Mastering Cloud Computing*, McGraw Hill 2017
3. Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, *Cloud Computing: A Practical Approach*, McGraw Hill 2017

**ICT 4046: DEEP LEARNING [3 0 0 3]**

**Objectives:**

* To introduce the mathematical tools for neural networks
* To provide exposure to the concepts of deep learning
* To provide exposure to TensorFlow for deep learning

**Abstract:**

Introduction, Mathematical Preliminaries, Machine Learning Basics, Deep Feedforward Networks, Regularization for Deep Learning, Optimization for Training Deep Models, Convolutional Networks, Recurrent and Recursive Networks, Practical Methodology

**Syllabus:**

**Introduction:**

Limitations of neural networks, Trends in Deep Learning [**1 Hour**]

**Mathematical Preliminaries**:

Linear Algebra, Probability and Information Theory, Numerical computation [**4 Hours**]

**Machine Learning Basics**:

Learning Algorithms, Capacity, Under and Overfitting, Hyperparameter and Validation Set, Estimators, Bias and Variance, MLE, Bayesian Statistics, Supervised Learning Algorithms, Unsupervised Learning Algorithms [**2 Hours**]

**Deep Feedforward Networks**:

Learning XOR, Gradient-Based Learning, Hidden Units, Architecture Design, Back-Propagation Algorithm [**5 Hours**]

**Regularization for Deep Learning**:

Parameter Norm Penalties, Norm Penalties as Constrained Optimization, Regularization and Under-Constrained Problems, Dataset Augmentation, Noise-Robustness, Semi-Supervised Learning, Multi-Task Learning, Early Stopping, Parameter Tying and Parameter Sharing, Sparse Representations, Bagging and Other Ensemble Methods, Dropout, Adversarial Training

[**6 Hours**]

**Optimization for Training Deep Models**:

Challenges in Neural Network Optimization, Basic Algorithms, Parameter Initialization Strategies, Algorithms with Adaptive Learning Rates, Approximate Second-Order Methods, Optimization Strategies and Meta-Algorithms [**5 Hours**]

**Convolutional Networks**:

Convolution Operation, Pooling, Convolution and Pooling, Variants of Convolution Function, Structured Outputs, Data Types, Efficient Convolution Algorithms, Random or Unsupervised Features [**5 Hours**]

**Recurrent and Recursive Networks**:

Unfolding Computational Graphs, Recurrent Neural Networks, Bidirectional RNNs, Encoder-Decoder Sequence-to-Sequence Architecture, Deep Recurrent Networks, Recursive Neural Networks, Echo State Networks, LSTM [**6 Hours**]

**Practical Methodology**:

Performance Metrics, Default Baseline Models, Selecting Hyperparameters, Debugging Strategies [**2 Hours**]

**Outcomes:**

By the end of this course, student should be able to:

* Choose right deep learning model and architecture for a given learning problem
* Implement deep learning architecture on platforms like Caffe, Theano or TensorFlow
* Comprehend and communicate the content of a research paper on deep learning

**References:**

1. Goodfellow I., Bengio Y., and Courville A., *Deep Learning*, MIT Press 2017.
2. Haykin S., *Neural Networks and Learning Machines*, PHI, 2016.
3. Patterson J., and Gibson A., *Deep Learning: A Practitioner's Approach, O'Reilly*, 2017.

**ICT 4047: GAME THEORY WITH ENGINNERING APPLICATIONS [3 0 0 3]**

**Objectives:**

* To provide the comprehensive treatment of non-cooperative game theory
* To provide the comprehensive treatment of cooperative game theory
* To introduce the fundamental concepts of mechanism design
* To introduce game-theoretic approach to solve engineering problems

**Abstract:**

Introduction to Game Theory and Mechanism Design, Mathematical Preliminaries, Non-Cooperative Game Theory, Cooperative Game Theory, Mechanism Design

**Syllabus:**

**Introduction:**

Game theory-the science of strategic interactions, current trends and applications [**1 Hour**]

**Mathematical Preliminaries:**

Probability theory, Linear algebra, Linear programming and duality, Mathematical analysis, Computational complexity classes [**2 Hours**]

**Non-Cooperative Game Theory:**

Key notations in game theory, Extensive form games, Strategic form games, Dominant strategy equilibria, Pure strategy Nash equilibria, Mixed strategies and mixed strategy Nash equilibria, Matrix games, Bayesian games [**14 Hours**]

**Cooperative Game Theory:**

Correlated strategies and correlated equilibrium, Two person bargaining problem, Coalition games with transferable utility, Core of coalition games, Shapley values, Other solution concepts in cooperative game theory, Stable matching [**13 Hours**]

**Mechanism Design:**

Introduction to mechanism design, Social choice functions, Incentive compatibility and revelation theorem, Auctions [**6 Hours**]

**Outcomes:**

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

* Identify strategic situations and represent them as games
* Solve games using various techniques
* Identify and prescribe strategies to implement for a given problem
* Analyse and solve engineering problems using game theoretic techniques

**References:**

1. Narahari Y., *Game Theory and Mechanism Design*, World Scientific, Chennai, 2015.
2. Fudenberg D, and Tirole J., *Game Theory*, ANE Books, New Delhi, 2015.
3. Bauso D.*, Game Theoy with Engineering Applications*, SIAM, Philadelphia, 2016.
4. Roughgarden T., *Twenty Lectures on Algorithmic Game Theory*, Cambridge University Press, 2016.

**ICT 4048: HIGH PERFORMANCE COMPUTING [3 0 0 3]**

**Objectives:**

* To describe the micro-architecture of CPU and GPU.
* To illustrate the fundamental concepts of parallel computing using GPU with CUDA toolkit.
* To demonstrate various data parallel algorithm primitives and optimization techniques.
* To analyze the performance of parallel programs.

**Abstract:**

Introduction to CPU & GPU micro-architecture, Flynn’s taxonomy, Amdahl’s law, Challenges in parallel programming, Parallel programming languages, Introduction to CUDA C programming, Data parallelism, Mapping threads to multidimensional data, Synchronization, Querying device properties, Data Parallel Algorithm Primitives, Reduction, Parallel histogram computation, Convolution, Parallel prefix sum, Introduction to CUDA Library, Basic thrust features, Interoperability, Thrust algorithm, Optimization Techniques, Memory optimizations, Common compiler optimizations, Profiling, Application Case Study.

**Syllabus:**

Introduction to CPU & GPU micro-architecture: Introduction to CPU micro-architecture: Instruction front end pipeline, cache hierarchy and cache coherency protocol.Introduction to GPU micro-architecture: Flynn’s taxonomy, Amdahl’s law, architecture of a modern GPU, Challenges in parallel programming, parallel programming languages.  [**4 HOURS**]

**Introduction to CUDA C Programming:**Data parallel computing: Data parallelism, CUDA C program structure, Simple programs involving single dimensional input data. Scalable parallel execution: CUDA thread organization, Mapping threads to multidimensional data, Synchronization, Resource assignment, Querying device properties, Thread scheduling and Latency tolerance, Programs involving multi-dimensional data. Memory and Data Locality: Importance of memory access efficiency, Matrix multiplication, CUDA memory types, Tiling for reduced memory traffic, Tiled matrix multiplication kernel, Programs on tiled approach.

**[10 HOURS]**

**Data Parallel Algorithm Primitives**: Reduction, Parallel histogram computation, Convolution, Parallel prefix sum, Programs involving application of primitives to perform scientific computation.   **[12 HOURS]**

**Introduction to CUDA Library:** Basic thrust features, Interoperability, Thrust algorithms**.**

**[4 HOURS]**

**Optimization Techniques**: Memory optimizations, Common compiler optimizations, Profiling, execution configuration optimizations. **[4 HOURS]**

**Application Case Study**: Machine Learning: Introduction to Convolutional Neural Networks and basic CUDA implementation. **[2 HOURS]**

**Outcomes:**

After completing the course the students will be able to:

* Recognize the architectural differences between CPU and GPU.
* Illustrate the basic concepts of parallel computing on the GPU using the CUDA toolkit.
* Resolve the computational bottlenecks in the parallel programs using optimization techniques.
* Select the appropriate data parallel primitives to perform scientific computation.

**References**

1. Kirk D. B. and Hwu W., *Programming Massively Parallel Processors: A Hands-on Approach (3e)*, Morgan Kaufmann Publishers Inc., 2016.
2. Barlas G., *Multicore and GPU Programming: An Integrated Approach*. Morgan Kaufmann Publishers Inc., 2015.
3. Gaster B., Howes L., Kaeli D. R., Mistry P., and Schaa D., *Heterogeneous Computing with OpenCL (2e)*,  Morgan Kaufmann Publishers Inc., 2012.

**ICT 4049: HUMAN COMPUTER INTERACTION [3 0 0 3]**

Contexts for HCI, Processes for user-centered development, Different measures for evaluation, Usability heuristics and the principles of usability testing, Physical capabilities that inform interaction design, Cognitive models that inform interaction design, Social models that inform interaction design, Principles of good design and good designers, Accessibility, Interfaces for differently-aged population groups.

**Syllabus:**

**Contexts for HCI**:

The Human-Input Output Channels, Human Memory, Thinking, Psychology and the design of interactive systems, The Computer-Text entry devices, Positioning, pointing and drawing, Display devices, Physical controls, sensors and special devices, Memory, Processing and networks

**[4 Hours]**

**Processes for user-centered development**:

Organizational Design to Support Usability, The Four Pillars of Design, Development Methodologies, Ethnographic Observation, Participatory Design, Scenario Development, Social Impact Statement for Early Design Review **[4 Hours]**

**Different measures for evaluation**:

Goals of evaluation, Evaluation through expert analysis, Evaluation through user participation, Choosing an evaluation method **[4 Hours]**

**Usability heuristics and the principles of usability testing**:

Different types of interviews and questionnaires, Data collection and analysis, Questionnaire design, Heuristic evaluation, strengths and limitations of the techniques and selection of appropriate ones **[4 Hours]**

**Interaction design**:

The process of design, User focus, Scenarios, Navigation design, Screen design and layout, Iteration and prototyping **[4 Hours]**

**Cognitive models**:

Goal and task hierarchies, Linguistic models, The challenge of display-based systems, Physical and device models, Cognitive architectures **[4 Hours]**

**Social models that inform interaction design**: Organizational issues, Capturing requirements, Face-to-face communication, Conversation, Text-based communication, Group working

**[4 Hours]**

**Principles of good design and good designers:**

Principles to support usability, Standards, Guidelines, Golden rules and heuristics, HCI patterns

**[4 Hours]**

Accessibility, Interfaces for differently-aged population groups **[4 Hours]**

**Outcomes:**

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

* Develop conceptual vocabulary for analyzing human interaction with software: affordance, conceptual model, feedback, so forth.
* Use a conceptual vocabulary for analyzing human interaction with software
* Define a user-centered design process that explicitly takes account of the fact that the user is not like the developer or their acquaintances.
* Create and conduct a simple usability test for an existing software application.
* Comprehend the content of a research paper on HCI.

**References:**

1. Alan Dix, Janet E. Finlay, Gregory D. Abowd, and Russell Beale, Human-Computer Interaction, 3rd edition,Prentice Hall, 2003.
2. Ben Shneiderman, Catherine Plaisant, Maxine Cohen and Steven Jacobs, Designing the User Interface: Strategies for Effective Human-Computer Interaction, 5th edition, Addison-Wesely, 2009.
3. Jeffrey Rubin and Dana Chisnell. Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. 2nd Edition. New York: Wiley, 2008.
4. Yvonne Rogers, Helen Sharp and Jenny Preece, Interaction Design: Beyond Human - Computer Interaction, 3rd Edition, Wiley, 2011

## **ICT 4050: INTERNET OF THINGS [3 0 0 3]**

**Objectives:**

* To understand the principles of internetworking of embedded devices.
* To learn the state-of-art architectures for IoT.
* To understand various technologies and protocols aimed at enabling the formation of highly distributed and ubiquitous networks of seamlessly connected heterogeneous devices which can be fully integrated into the current Internet.
* To analyze and visualize sensor data

**Abstract:**

Introduction to M2M communication and IoT , An emerging industrial structure for IoT, IoT system architecture, IoT reference model, IoT deployment and operational view, IoT physical devices and endpoints, Communication and networking protocols-MQTT and AMQP protocols, IoT enabling technologies-RFID, WSN,SCADA etc., Analytics for the IoT, Applying the geospatial analytics to IoT data, Real world design constraint, Technical design constraint, Future internet design for various IoT use cases such as smart cities, smart environments, smart homes, smart health etc.

**Syllabus:**

**Introduction:**

M2M Communication, IoT, M2M value chain, IoT value chain, an emerging industrial structure for IoT, Implications for IoT, Barriers and concern, IoT use case example. **[3 Hours]**

**M2M to IoT – An Architectural Overview:**

An IoT architecture outline, Standards considerations. IoT data Management, IoT architecture-State of art solution, IoT reference model, IoT deployment and operational view.  **[7 Hours]**

**IoT Physical devices and endpoints:**

Basic building blocks of an IoT Device, Exemplary Device: Raspberry Pi, interfaces, Programming Raspberry Pi with Python. IoT physical servers and cloud offerings: introduction to cloud storage models and communication Networks, REST APIs along with HTTP, MQTT and AMQP protocols **[6 Hours]**

**IoT Enabling Technologies:**

**M2M**: The Internet of Devices, RFID: The Internet of Objects, WSN: The Internet of Transducers, SCADA: The Internet of Controllers. Web of Things versus Internet of Things, M2M and WSN Protocols, SCADA and RFID Protocols, Issues with IoT Standardization, Unified Data Standards **[10 Hours]**

**Analytics for the IoT:**

Data flows from the IoT device to the final data set, Develop techniques to wring value from IoT data, apply geospatial analytics to IoT data, Use machine learning as a predictive method on IoT data. . **[6 Hours]**

**Real-world Design Constraint:**

Technical design constraints, IoT devices and networks, data representation and visualization, interaction and remote control. **[2 Hours]**

**IoT Use Cases:**

Ubiquitous IoT Applications, Telematics and Intelligent Transport Systems, Smart Grid and Electric Vehicles, Smarter Planet and Smart Buildings,Home Healthcare and Remote Patient Monitoring.  **[2 Hours]**

**Outcomes:**

**Upon completion of the course, the students will be able to:**

* Demonstrate an internetwork between embedded devices through the Internet
* Apply the concept of IoT for a particular sensor based network
* Choose appropriate network architecture for a particular application.
* Analyse and design networks to support the development of intelligent services with given performance requirements in a variety of application domains.
* Assess different Internet of Things technologies and their applications.

References:

1. Holler J., Tsiatsis V., Mulligan C., Karnouskos S., Boyle D., *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence* *(1e),* Elsevier 2014.
2. Bahga A., Madisetti V., *Internet of Things-A Hands on Approach (1e),* Orient Blackswan Private Limited, 2015.
3. Roderick O., Marko N., Sanchez D. and Aryasomajula A., *Internet of Things and Data Analytics Handbook (1e),* Wiley-Blackwell, 2017.
4. Patil Y., *Azure IoT Development Cookbook (1e),* Packt publishing Ltd, 2017.
5. Minteer A., *Analytics for the Internet of Things (1e),* Packt publishing Ltd, 2017.

**ICT 4051: NATURAL COMPUTING [3 0 0 3]**

**Course Objectives:**

1. To understand various existing computing in nature.
2. To give the student an introduction to various algorithms in the area of natural computing

and show how they have proven to be very powerful in solving various kinds of problems.

1. To have basic proficiency in using softwares such as Xgrow,Xtile,ISUTAS,CadNano,Sarse,Tiamat,Chemical Compiler etc

**Abstract:**

Basic Notations of Biochemistry and Molecular Biology, DNA Computing, Basic Computing Models: Finite Automata (FA), Push Down Automata (PDA), Linear Bounded Automata (LBA) and Turing Machine (TM), Quantum Turing Machine (QTM) and Quantum Languages, Computation by circuits, Thermodynamics of Computation, Algorithmic Botany, Cellular Automata, DNA Computation Models:Lipton Model, Sticker model, DNA Splicing model, DNA Self Assembly, Hairpin Model, Algorithms for Natural Security and Cryptography, Experiments in Self-Assembly, DNA Origami (2D and 3D), Error-Correction in Self- Assembly, Bacterial Computers and Data Storage, Peptide Computing, Membrane Computing, Chemical Computing.

**Syllabus :**

Basic Notations of Biochemistry and Molecular Biology [2 hours]

DNA Computing: Introduction, Encoding scheme, Comparison with conventional computing, Application of DNA computing [4 hours]

Basic Computing Models: Finite Automata (FA), Push Down Automata (PDA), Linear Bounded Automata (LBA) and Turing Machine (TM), Quantum Turing Machine (QTM) and Quantum Languages, Computation by circuits, Thermodynamics of Computation, Algorithmic Botany, Cellular Automata [10 hours]

DNA Computation Models :Lipton Model, Sticker model, DNA Splicing model, DNA Self Assembly, Hairpin Model, Algorithms for Natural Security and Cryptography, Experiments in Self-Assembly, DNA Origami (2D and 3D), Error-Correction in Self- Assembly [10 hours]

Bacterial Computers and Data Storage, Peptide Computing, Membrane Computing, Chemical Computing [10 hours]

**Course Outcomes:**

**Upon completion of the course, the students will be able to:**

1. Appreciate the concepts and complexity behind natural computing.
2. Use various natural computing algorithms to other domains of computing.
3. Use natural computing algorithm based software to understand the concept behind natural computing and apply these programs to other domains of computing

**References:**

1. Leandro Nunes de Castro, *Fundamentals of Natural Computing: Basic Concepts, Algorithms and Applications*, CRC Press, USA, 2006.
2. Ignatova Z., Martnez-Prez I., and Karl-Heinz Zimmermann, *DNA Computing Models*, Springer, 2008.
3. Amos M., *Theoretical and Experimental DNA Computation*, Springer, 2005.
4. Hopcroft J.E., Motwani R., and Ullman J.D., *Introduction to Automata Theory, Languages and Computation*, Pearson Asia, 2001.

**ICT 4052: NEURAL NETWORKS AND FUZZY LOGIC [3 0 0 3]**

**Course Objectives:**

* To have an understanding for the engineering issues underlying the design of neural networks and fuzzy logic-based systems.
* To introduce the conceptual and mathematical foundation of neural networks and fuzzy logic.
* To have basic proficiency in using Neural Network and Fuzzy Logic Toolbox in MATLAB/Python.

**Abstract**

Introduction to Neural Networks and Fuzzy Logic, Learning Processes, Single-Layer Perceptron, Multi-Layer Perceptron, Radial Basis Function Networks, Support Vector Machine, Fuzzy Sets and Crisp Sets, Fuzzy Relations, Membership Functions, Fuzzy Logic and Inference, Membership Development

**Syllabus**

**Introduction:**

Human Brain, Models of a neuron, Neural networks as directed graphs, network architecture [**2 Hours**]

**Learning Processes:**

Error-correction learning, Memory-based learning, Hebbian learning, Competitive learning, Boltzmann learning, Learning with a teacher, Learning without a teacher, Learning task, memory [**4 Hours**]

**Single Layer Perceptron:**

Adaptive filtering problem, Unconstrained optimization techniques, Linear least-square filters, Least-mean square algorithm, Learning curves, Learning rate annealing, Perceptron, Perceptron convergence theorem [**4 Hours**]

**Multilayer Perceptron:**

Back-propagation algorithm, XOR problem, Heuristics for improving the performance of back-propagation algorithm. [**4 Hours**]

**Radial Basis Function Networks:**

Cover’s theorem, Interpolation problem, Supervised learning as ill-posed problem, Regularization theory, Regularization networks, Generalized radial-basis function networks, XOR problem [**4 Hours**]

**Support Vector Machine:**

Optimal hyperplane for linear separable patterns, Optimal hyperplane for non-separable patterns, Building a SVM for pattern recognition, Epsilon-Insensitive loss function, SVM for non-linear regression [**5 Hours**]

**Introduction to Fuzzy Sets and Crisp Sets** [**1 Hour**]

**Fuzzy Relations:**

Crisp relations, Fuzzy relations, Tolerance and equivalence relations, Fuzzy tolerance and equivalence relations [**2 Hours**]

**Membership Functions:**

Standard forms and boundaries, Fuzzyfication, Membership value assignment [**2 Hours**]

**Fuzzy Logic and Inference:**

Fuzzy logic, Fuzzy Systems-Natural Language, Linguistic Hedges, Fuzzy Rule based Systems

[**4 Hours**]

**Membership Function Development:**

Membership value assignments-Intuition, Inference, Rank ordering, Neural networks, Genetic algorithms, Inductive reasoning [**4 Hours**]

**Learning Outcomes:**

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

* Understand the conceptual issues behind the working of neural networks and fuzzy logic based systems.
* Use various mathematical tools to design neural and fuzzy logic-based system for the given application.
* Write MATLAB/Python programs using Neural Network Toolbox and Fuzzy Logic Toolbox to implement the designed NFL systems.

**References:**

1. Haykin S.*, Neural Networks and Learning Machine*, *(3e),* Pearson Education, New Delhi, 2016.
2. Hagan M.T., Demuth H B., and Beale M.H., *Neural Network Design*, *(2e),* Pearson Education, New Delhi, 2014.
3. Ross T.J., Fuzzy Logic with Engineering Applications*, (3e),* Wiley, USA, 2011.

**ICT 4053: PATTERN RECOGNITION [3 0 0 3]**

**Objectives:**

* To understand the concepts of Bayesian theory.
* To comprehend the features for decision making process.
* To analyse the algorithm for classifying and clustering the data

**Abstract:**

Machine perceptron, Pattern recognition, design cycle, Minimum error rate classification, Discriminant features, Normal desnsity, Bayesian belief network, Missing and noisy features, Maximum-likelihood estimation, Gaussian case, Computational complexity, Fisher linear discriminant, Nearest neighbor, Fuzzy classification, Linear discriminant function, Minimizing Perceptron criterion function, Relaxation MSE, Ho-Kashyap procedures, Hidden Markov models, Evaluation, Decoding, Learning, Unsupervised bayesian learning, Criterion functions for clustering, Iterative optimization, Hierarchical clustering

**Syllabus:**

**Introduction to Pattern Classification and Structural Pattern Recognition**:

Machine perception, Pattern recognition systems, Design cycle, Learning and adaptation.

**[2 Hours]**

**Bayesian Decision Theory**:

Continuous features, Minimum error rate classification, Discriminant functions and decision surfaces, Normal density, Discrete features, Missing and noisy features, Bayesian belief networks. **[6 Hours]**

**Bayesian Parameter Estimation and Dimensionality Problem:**

Maximum-Likelihood estimation, Bayesian estimation, Gaussian case and General theory, Problem of dimensionality: Dimension and accuracy, Computational complexity, Fisher linear discriminant, Multiple linear discriminant **[7 Hours]**

**Nonparametric Pattern Recognition:**

Density estimation, Parzen windows, K-nearest-neighbor estimation, Nearest-neighbor rule, Fuzzy classification. **[6 Hours]**

**Linear Discriminant Function:**

Linear discriminant functions and decision surface, Generalized linear discriminant functions, Two-category linearly separable case, Minimizing perceptron criterion functions, Relaxation procedures, MSE procedures, Ho-Kashyap procedures.  **[6 Hours]**

**Time Varying Pattern Recognition:**

First order hidden markov models, Hidden markov model computation, Evaluation, Decoding, Learning **[4 Hours]**

**Unsupervised Classification and Clustering**: Unsupervised bayesian learning, Criterion functions for clustering, Iterative optimization, Hierarchical clustering. **[5 Hours]**

**Outcomes:**

By the end of this course, the students are able to:

* Apply relevant mathematical or programming tools using Bayesian theory to solve problems for estimation and dimensionality reduction
* Solve the problems based on time, non-parametric pattern recognition and discriminant functions.
* Analyse the various algorithms for their effective usage in classification and clustering

**References:**

1. Duda R. O., Peter E.H. and David G. S., *Pattern Classification* *(2e)*, Wiley-Interscience, 2000.
2. Fukunaga K., *Introduction to Statistical Pattern Recognition* *(2e),* Academic ress, 2013.
3. Bishop C. M., *Pattern Recognition and Machine Learning* *(1e),* Springer, 2011.
4. Hastie T., Tibshirani R.and Friedman J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* *(2e*), Springer, 2017.

## **ICT 4054: SOCIAL NETWORK ANALYSIS [3 0 0 3]**

**Objectives:**

* To describe social network models with different metrics
* To illustrate the characteristics of social network.
* To analyse structural and population models
* To recall visualization technique using Gephi.

**Abstract:**

Introduction to social networks: matrices used to analyze the network, types of network mode, Graph Concepts: connectivity, DFS and BFS in network, Datasets, Strong and Weak Ties: Strength of weak ties, Network structure in Large scale data, Graph Partitioning, Networks in their surrounding contexts: Homophily, Spatial model of Segregation, Positive and negative relations: Structural balance, characteristics, applications Information cascades: Baye’s rule, Simple cascade model, Network effects: Economy with and without network effects, Power laws and Rich Get Richer Phenomena: Effect of search tools and recommendation system, analysis of Rich-Get-Richer Processes, Cascading behavior in the network: Diffusion, cascades and clusters, Small world phenomena: Six Degrees of Separation, Decentralized Search, Visualization using Gephi: Graph layout algorithm

**Syllabus:**

**Becoming familiar with social networks**

Describing social network through graphs and graph theory, Network Matrices, One mode and two mode networks, How to study social networks from theory to Design **[3 Hours]**

**Reference : 2**

**Graphs**

Basic Definitions, Paths and Connectivity, Distance and Breadth First Search, Network Data sets: An Overview **[2 Hours]**

**Reference : 1**

**Strong and Weak Ties**

Triadic Closure, The Strength of Weak Ties, Tie Strength and Network Structure in Large Scale data, Tie, Strength, Social Media and Passive Engagement, Closure, Structural Holes and Social capital, Betweenness Measures and Graph Partitioning **[4 Hours]**

**Reference : 1**

**Networks in Their Surrounding Contexts**

Homophily, Mechanisms Underlying Homophily : Selection and Social Influence, Affiliation, Tracking Link Formation in Online Data, A Spatial Model of Segregation  **[3 Hours]**

**Reference : 1**

**Positive and Negative Relations**

Structural Balance, Characterizing the Structure of Balanced Networks, Applications of Structural Balance, A Weaker from of Structural balance, Advanced material : Generalizing the Definition of Structural Balance **[3 Hours]**

**Reference : 1**

**Network Dynamics: Population Models**

**Information Cascades**

Following the Crowd, A Simple Herding Example, Bayes’ Rule, Bayes’ Rule in Herding Experiment, A simple general cascade model, Sequential Decision Making and cascades

**[3 Hours]**

**Reference : 1**

**Network Effects**

The Economy without Network Effects, The Economy with Network Effects, Stability, Instability, and Tipping Points, A Dynamic View of the market, Industries with Network Goods, Mixing Individual Effects with Population Level Effects, Negative Externalities and El Farol Bar Problem

**[4 Hours]**

**Reference : 1**

**Power Laws and Rich Get Richer Phenomena**

Popularity as a Network Phenomenon, Power Laws, Rich-Get-Richer Models, The Unpredictability of Rich-Get-Richer Effects, The Long Tail, The effect of Search Tools and Recommendation Systems, Analysis of Rich-Get-Richer Processes **[3 Hours]**

**Reference : 1**

**Network Dynamics : Structural Model**

**Cascading Behavior in Networks**

Diffusion in Networks, Modeling Diffusion through a Network, Cascades and Clusters, Diffusion, Thresholds and the Role of Weak Ties, Extensions of the Basic Cascade Model, Knowledge, Thresholds and Collection Action, The cascade capacity **[3 Hours]**

**Reference : 1**

**The Small World Phenomenon**

Six Degrees of Separation, Structure and Randomness, Decentralized Search, Modeling the Process of Decentralized Search, Empirical Analysis and Generalized Models, Core-Periphery Structures and Difficulties in Decentralized Search, Analysis of Decentralized Search **[4 Hours]**

**Reference : 1**

**Network Visualization**

Basic Graph Manipulations, Using graph layout algorithm **[4 Hours]**

**Reference : 4**

**Outcomes:**

By the end of this course, the student are able to:

* Define various models of social network with different metrics
* Correlate the characteristics of network
* Identify the network effects taking place in a given context
* State the role of nature of ties and social context in influencing a network
* Analyse the network data to perform various computations on datasets

**References:**

1. Easley D. and Kleinberg J., *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press, 2010.
2. Prell C., *Social Network Analysis*, SAGE Publications, 2012.
3. Yang S., Keller F. B., Lu Zheng, *Social Network Analysis*, SAGE Publications, 2017.
4. Khokhar D., *Gephi Cookbook*, Packt Publishing, 2015.

**ICT 4055: SOFTWARE RELIABILITY [3 0 0 3]**

**Objectives:**

* To understand scientific concepts of Software and Hardware Reliability
* To learn the application of Software Reliability Models
* To learn software metrics that define relevant metrics in a rigorous way.

**Abstract:**

Need and concepts of software reliability; Software reliability models-classification, limitations and issues; model disagreement and inaccuracy, predictive accuracy, recalibration; The operational profile –concepts and development procedures, test selection; Testing for reliability measurement; Software testing; operational profiles – difficulties, estimating reliability, time/structure based software reliability; Fundamentals of measurement product metrics **–**measurement of internet product attributes, size and structure , measurement of quality; Reliability growth model.

**Syllabus:**

Need and Concepts of Software Reliability

**Failure and Fault**:

Prevention, Removal, Tolerance, Forecast, Dependability Concept– Failure Behavior, Characteristics, Maintenance Policy, Reliability and Availability Modeling, Reliability Evaluation **[4 Hours]**

**Software Reliability Models:**

Introduction - Historical Perspective and Implementation, classification, limitations and issues, Exponential Failure Models – Jelinski-moranda model, Poisson, Musa, Exponential models, Weibull Model, Musa-okumoto Model, Bayseian Model –Littlewoodverral Model, Phase Based Model **[9 Hours]**

**Prediction Analysis:**

Model Disagreement and Inaccuracy – Short & Long Term Prediction, Model Accuracy, Analyzing Predictive Accuracy – Outcomes, PLR, U & Y Plot, Errorsand Inaccuracy, Recalibration – Detecting Bias, Techniques, Power ofRecalibration, Limitations in Present Techniques, Improvements.  **[6 Hours]**

**The Operational Profile:**

Concepts and Development Procedures – Customer Type, User Type, SystemMode, Functional and Operational Profile, Test Selection - Selecting Operations, Regression Test. **[3 Hours]**

**Testing For Reliability Measurement:**

Software Testing – Types, White and Black Box, Operational Profiles – Difficulties, Estimating Reliability, Time/Structure based software reliability – Assumptions, Testing methods, Limits, Starvation , Coverage, Filtering, Microscopic Model of Software Risk. **[6 Hours]**

**Fundamentals of Measurement:**

Measurements in Software Engineering – Scope of Software metrics – Measurements theory – Goal based Framework – Software Measurement Validation. **[4 Hours]**

**Product Metrics:**

Measurement of Internet Product Attributes – Size and Structure – External Product Attributes – Measurement of Quality –Reliability Growth Model – Model Evaluation **[4 Hours]**

**Outcomes:**

**Upon completion of the course, the students will be able to**

* Understand the fundamentals of measurement in software engineering.
* Analyse the need of software metrics for quality control and assurance.
* Apply Software Reliability Growth Models in Software Development
* Perform simple statistical analysis relevant to software measurement data.

**References:**

1. Lyu M., *Handbook of Software Reliability Engineering*, IEEE Computer Society Press, ISBN: 0-07-039400-8, 1996.
2. Connor P. D., *Practical Reliability Engineering* *(5e)*, John Wesley & sons, 2012.
3. Fenton N. E., Beiman J., *Software metrics-A rigorous and practical approach* *(3e)*, Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series, 2014.
4. Musa J. D., *Software Reliability Engineering* *(2e)*, Tata McGraw Hill, 2005.
5. Yamada S., *Software Reliability Modeling: Fundamentals and Applications (Springer Briefs in Statistics)*, 2014 Edition, Springer, 2014.

**OPEN ELECTIVES**

**ICT 4301: COMPUTER GRAPHICS AND ANIMATION [3 0 0 3]**

**Objectives:**

* Describe the visual communications developed over history
* Discuss basic design procedures, transformations, projections
* Design computer graphics algorithm.
* Understand the rationale, issues, and directions in the development of computer graphics

**Abstract:**

Introduction to Computer Graphics, Video Display Devices, Raster Scan Systems, Graphics Output Primitives, Line Drawing Algorithms, Circles and Ellipses Generating Algorithms, Geometrical Transformation, Two Dimensional and Three Dimensional Transformations, Inverse Transformations, Three Dimensional Translation, Rotation and Scaling, , Transformation, Two Dimensional and Three Dimensional Viewing , Animation, Raster Methods for Computer Animation, Design of Animation sequences, Articulated Figure Animation, Periodic Motion, Graphics Programming using OpenGL,

**Syllabus:**

**Overview Of Graphics Systems:**

Video display devices, Raster scan systems, Graphics software, Introduction to OpenGL.

**[4 Hours]**

**Graphics Output Primitives:**

Line drawing algorithms, Circles and Ellipses generating algorithms, General Scan line polygon fill algorithm, Scan line fill of convex polygons and regions with curved boundaries, filling of areas with irregular boundaries.  **[9 Hours]**

**Geometrical Transformations:**

Basic 2D transformations, Matrix representation and Homogeneous coordinates, Inverse transformations, 2D composite transformations, Geometric transformations in 3D space, 3D translation, rotation and scaling, Composite 3D transformations **[7 Hours]**

**Two Dimensional and Three Dimensional Viewing**

2D viewing Pipeline, Clipping window, Normalization and viewport transformations, clipping algorithms, Overview, 3D viewing pipeline, Projection transformations, Orthographic projections, Oblique parallel projections, Perspective projections. **[5 Hours]**

**Animation:**

Raster methods for computer animation, Design of animation sequences, Traditional and Computer animation techniques, Key-Frame systems, Motion specifications, Articulated figure animation, Periodic motion. **[6 Hours]**

**Graphics Programming using openGL:**

Why OpenGL, Features in OpenGL, OpenGL operations, Abstractions in OpenGL – GL, GLU & GLUT, a few examples and demos of OpenGL programs.[**5 Hours**]

**Course Outcomes**

**At the end of the course students are able to:**

* Explain the basic concepts of Computer Graphics
* Develop new algorithms for various transformations
* Review current status, research directions in the field of computer graphics
* Apply the concept of computer graphics to game design as well as information visualization

**References**

1. Donald D. Hearn, Warren Carithers, M. Pauline Baker. *Computer Graphics with OpenGL*(4ed), Pearson, Education, 2014.
2. Zhigang Xiang, *Computer Graphics: Theory and Practice with OpenGL* (3ed), Pearson Education, 2016.
3. Edward Angel, *Interactive Computer Graphics- A top down approach using OpenGL* (5ed), Pearson Education, 2012
4. Foley J. D., VanDam A., Feiner S. K., Hughes J. F., *Computer Graphics, Principles and Practice* (3ed), Addision-Wesley, 2014.

**ICT 4302: DESIGN AND DEVELOPMENT OF WEB APPLICATIONS [3 0 0 3]**

**Objectives:**

* Understand basics of internet and web
* Understand various tools and languages required for web design.
* Ability to build web pages and website.

**Abstract:**

Introduction to Internet and Web. Basic components of web page and fundamental concepts of languages required for Client side scripting and Server side scripting: HTML markup for structure of the simple page, text, creating links, adding images, table markup and forms. CSS for presentation: Orientation, formatting text, different selectors, colors background plus even more selectors, box model, padding, border, positioning, layouts, transitions, transformation and animation. Java script data types, control structure, DOM, arrays, objects, event handling. PHP introduction, control statements, arrays, objects, functions, connecting to mysql data base.

**Syllabus:**

**Introduction:**

How web works: An explanation of the web as it relates to the internet. Role of browser and server. Introduction to URL and its components, anatomy of a web page. **[1 Hour]**

**HTML:**

HTML markup for structure: Creating a simple page, marking text, creating links, adding images, table markup, forms**. [7 Hours]**

**CSS:**

CSS for Presentation: Cascading Style Sheets Orientation, Formatting Text Plus More Selectors Colors and Backgrounds Plus Even More Selectors and External Style Sheets. Thinking Inside the Box Padding, Borders, and Margins Floating and Positioning Page Layout with CSS, Transitions, Transforms, and Animation, CSS Techniques. **[8 Hours]**

**JAVASCRIPT:**

Introduction to Javascript decision statements, control statements.Using Javascript, DOM, functions, arrays, objects, talking to the page by event handling, getting valid input. **[10 Hours]**

**PHP:**

Introduction: Introduction to PHP, expressions and control flow in PHP, arrays, functions, objects etc. connecting to mysql database.  **[10 Hours]**

**Outcome:**

**By the end of the lab course, the students are able to**

* Describe the basics of web and internet.
* Develop web pages using HTML and CSS
* Validate the web pages using Javascript
* Add server side scripting through PHP.

**References:**

1. Robbins J.N., *Learning Web Design (4e)*, O’reilly Media, reprint 2017.
2. Harris A., *HTML 5 and CSS 3: All in One for Dummies (3e),* Wiley Brand, reprint 2017.
3. Nixon R., *Learning PHP, MySQL, JavaScript with Jquery CSS and HTML5 (4e*), O’reilly Media, reprint 2017.

**ICT 4303: FUNDAMENTALS OF DATA STRUCTURES AND ALGORITHMS [3 0 0 3]**

**Objectives:**

* Design efficient algorithms for various problems
* Understand the basic concepts of linear and nonlinear data structures.
* Compare and contrast various searching and sorting techniques
* To apply data structure concepts for efficient representation of data

**Abstract:**

Introduction to algorithms, Arrays: Elementary operations, Applications, Performance Analysis, Sparse matrix representation, Transpose of sparse matrix, Stacks operations, Arithmetic expression conversion and evaluation using stack, Queue Operations, Singly linked Lists, Circular lists, Doubly linked lists, Trees, Binary Tree traversals and different operations, Binary search Tree, Heaps, Graph Abstract type: Representations and elementary operations, Sorting and searching techniques, Analysis of algorithm.

**Syllabus:**

**Introduction:**

Performance Analysis and Measurements – Asymptotic notations, introduction to data structure, classification of data structure, Abstract data types **[4 Hours]**

**Arrays:**

The Array as Abstract Data type, Sparse Matrix Representation, Transpose of a sparse matrix, Representation of multidimensional arrays, The String abstract data type, Pattern matching.

**[3 Hours]**

**Stacks:**

Definition, operations on stacks, Evaluation of Arithmetic Expressions, Conversion of arithmetic expressions, Recursion, Multiple Stacks **[3 Hours]**

**Queues:**

Definition, operations, application of circular queues. **[2 Hours]**

**Linked Lists:**

Introduction to pointers and Dynamic memory allocation, Singly linked lists, Circular lists, Dynamically Linked Stacks and Queues, Polynomial representation and polynomial operations using singly linked list, Singly circular linked list, Doubly linked lists, Analysis of linked list operations. **[8 Hours]**

**Trees:**

Tree terminology, Binary trees, Properties, Binary tree representations, Binary Tree Traversal algorithms, Expression tree, Heaps, Binary Search Trees. Complexity associated with various algorithms. [**8 Hours]**

**Graphs:**

Definitions and Representations, Depth First Search, Breadth First Search, Connected components, Spanning trees, Complexities associated with each of the searching techniques.

**[4 Hours]**

**Sorting and Searching**:

Insertion Sort, Quick Sort, Merge sort, Heap sort, Shell sort, Linear search, Binary search, analysis of algorithms with respect to time complexity **[4 Hours]**

**Outcomes:**

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

* Summarize asymptotic notations to represent the complexities of the algorithms.
* Apply the appropriate data structure for the given problem.
* Compare the performance of sorting and searching techniques.
* Develop an efficient algorithm for the given problem

.

**References:**

1. Horowitz E., Sani S., Dinesh Mehta, *Fundamentals of Data Structures in C*++ *(2e),* Golgotha Publications, 2008.
2. Weiss M. A., *Data Structures and Algorithm Analysis in C++ (3e),* Pearson Education, 2009.
3. Horowitz E., Sahni S., Anderson-Freed S.*, Fundamentals of Data structures in C (2e),* Silicon Press, 2008.

**ICT 4304: MACHINE LEARNING TOOLS AND TECHNOLOGIES [3 0 0 3]**

**Objectives:**

* To introduce the generic principle of learning.
* To introduce the basic principles, techniques, and applications of Machine Learning.
* To have proficiency in Python to write programs for Machine Learning algorithm

**Abstract:**

Introduction to Machine Learning, Mathematical Preliminaries, Supervised Learning-LMS, logistic regression, GDA, Naive Bayes, SVM, model selection, Learning theory-bias/variance tradeoff, union and Chernoff bounds, VC dimensions, Unsupervised learning-clustering, k-means, Gaussian mixture, factor analysis, PCA, ICA, Machine Learning with Python.

**Syllabus:**

**Introduction:**

Basic concepts of machine learning, some day to day examples of machine learning [**1 Hour**]

**Mathematical Preliminaries:**

Review of Linear Algebra, Probability Theory Review, Overview of Convex Optimization, Hidden Markov Models, Multivariate Gaussian Distribution, Gaussian Processes **[4 Hours]**

**Supervised Leaning:**

Supervised learning setup, LMS, Logistic regression, Perceptron, Exponential family, Generative learning algorithms, Gaussian discriminant analysis, Naive Bayes, Support vector machines, Model selection and feature selection, Ensemble methods: Bagging, boosting. Evaluating and debugging learning algorithms  **[12 Hours]**

**Learning Theory:**

Bias/variance tradeoff, Union and Chernoff and Hoeffding bounds, VC dimension, Worst case (online) learning, Practical advice on how to use learning algorithms [**4 Hours**]

**Unsupervised Learning:**

Clustering, K-means, EM, Mixture of Gaussians, Factor analysis, PCA (Principal components analysis), ICA (Independent components analysis) [**8 Hours**]

**Reinforcement Learning:**

Markov Decision Processes (MDPs), Bellman equations, Value iteration and policy iteration, Linear quadratic regulation (LQR), LQG, Q-learning, Value function approximation, Policy search, Reinforce, POMDPs [**7 Hours**]

**Course Outcomes:**

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

* Understand machine learning algorithms and their use in data-driven knowledge discovery.
* Identify, formulate and solve machine learning problems that arise in practical applications.
* Implement several machine learning algorithms in Python

**References:**

1. Murphy K.P., *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.
2. Mohri M., Rostamizadeh A., and Talwalkar A., *Foundations of Machine Learning*, MIT Press, 2012.
3. Koller D., and Friedman N, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, 2009.
4. Bishop C.M., *Pattern Recognition and Machine Learning, (2e),* Springer, 2013.

**ICT 4305: NETWORKING WITH TCP/IP [3 0 0 3]**

**Objectives:**

* To understand OSI and TCP/IP network models
* To discuss functionalities of different layers
* To understand protocols of different layers

**Abstract :**

Introduction to Networking and brief History of Internet, OSI and TCP/IP Reference Models, Network Layer, IP Addresses, Internet Protocol (IP) Datagram, Fragmentation, Options, Address Translation, ICMP and IGMP, Intra and Inter domain Routing, Distance Vector Routing, RIP, Link State Routing, OSPF, Path Vector Routing, BGP, User Datagram and Transmission Control Protocol, SCTP, Application Layer Protocols, The Web and HTTP, DNS: Services Provided by the DNS

**Syllabus:**

**Introduction to Computer Networks and layered Architecture:** Definition, Uses, Classification of Networks, Network topology and Topography, Layers, Protocols and services, ISO/OSI Reference Model, Overview of TCP/IP architecture, MAC, Application Protocols and TCP/IP utilities.**[3 Hours]**

**IP Addresses**:  Classful Addresses, Subnetting and supernetting and subnet supernet mask in classful, special addressing, Variable length blocks, subnetting and subnetmask in classless addressing**.                                                                                                                           [3 Hours]**

**Internet Protocol**: Datagram, Fragmentation, Options, Address Translation (NAT)  **[3 Hours]**

**ICMP and IGMP:** Internet Control Message Protocol **-**Types of protocol, Message format, Error reporting, Query. Internet Group Management Protocol - Group management, IGMP Messages, IGMP operation, Encapsulation                                                                              **[5 Hours]**

**Unicast Routing Protocol**: Intra and Inter domain Routing, Distance Vector Routing, RIP, Link State Routing, OSPF, Path Vector Routing, BGP                 **[4Hours]**

**User Datagram Protocol and Transmission Control Protocol:** Relationship between Transport and Network layer, Overview of Transport layer in the Internet, Process to Process Communication, User datagram Segment Structure, TCP services, TCP Features, Segment, TCP connection, State Transition diagram, Flow Control, Error control, Congestion Control, TCP Timers                     **[6 Hours]**

**Stream Control Transmission Protocol:** SCTP Services, SCTP features, Packet format, SCTP association, Flow control, Error Control, Congestion control        **[4 Hours]**

**Application Layer:** Principle of Network Applications, The Web and HTTP: Non persistent and Persistent connection, HTTP Message Format, User-Server Interaction: Cookies, HTTP content, Web Caching, The conditional GET,   FTP: FTP commands and replies, Electronic Mail in the Internet: SMTP, Comparison with HTTP, Mail Message Format and MIME, Mail Access Protocol, DNS: Services Provided by the DNS, Overview of How DNS works, DNS record and Messages, TELNET: Concept, NVT, NVT character set, Embedding, Options, Option Negotiation,  Suboption Negotiation, Mode of operation                                     **[8 Hours]**

**Course Outcomes:**

The student will be able to

* Identify the properties of OSI model and TCP/IP protocol suite.
* Illustrate the proper usage of the various protocols that has been used in the different layers of TCP/IP protocol suite.
* Differentiate the various services offered to the applications that use the Internet for audio and video services.

**References:**

1. Behrouz A. Forouzan, *TCP/IP Protocol Suite* (4e), Tata McGraw Hill 2017
2. Andrew S. Tanenbaum, *Computer Network* (5e), Prentice Hall of India Pvt Ltd 2013
3. James F. Kurose, *Computer Networking A top-Down Approach Featuring the Internet* (6e), Pearson Education Inc 2013.
4. Behrouz A. Forouzan, *Data Communications and Networking* (5e), Tata McGraw Hill 2013.

ICT 4306 CYBER SECURITY [3 0 0 3]

Objectives

* To provide introduction to the fundamental principles of cybersecurity.
* Describe the browser security model including same-origin policy and threat models in web security.
* Discuss the concept of web sessions and secure communication channels
* Understand cybercrime, cybercrime investigation
* Understand the laws and ethics of cybersecurity

Abstract :

Introduction to Information, Network and System Security, Encryption techniques, Message Integrity and Message Authentication, Digital Signature, Key Management, User Authentication. Web security model: Browser security model including same-origin policy, Client-server trust boundaries, Session management, authentication: Single sign-on, HTTPS and certificates. Application vulnerabilities and defenses: SQL injection, XSS, CSRF. Client-side security: Cookies security policy, HTTP security extensions, Plugins, extensions, and web apps, Web user tracking, Server-side security tools, e.g. Web Application Firewalls (WAFs) and fuzzers. Cybercrime, Cybercrime investigation, Laws and ethics

Syllabus:

Introduction: Information, network and system security [2 hours]

Introduction to cryptography: Introduction, Examples and applications in cybersecurity [4 hours]

Authentication and integrity: Integrity and Message Authentication, Digital Signatures, Hash functions. [4 hours]

Web security model : Browser security model including same-origin policy, Client-server trust boundaries [2 hours]

Session management, authentication: Single sign-on,HTTPS and certificates [2 hours]

Server side and Client side security: SQL injection, XSS,CSRF, Web Application Firewalls (WAFs) and fuzzers, Cookies security policy, HTTP security extensions, ,Plugins, extensions, and web apps,Web user tracking. [4 hour]

Cybercrimes -Introduction and Overview of Cyber Crime - Nature and Scope of Cyber Crime - Types of Cyber Crime [4 hour]

Computer Intrusions - Introduction - White collar Crimes - Viruses and Malicious Code - Internet Hacking and Cracking - Virus Attacks – Software Piracy - Intellectual Property - Mail Bombs - Exploitation - Stalking and Obscenity in Internet - Digital laws and legislation - Law Enforcement Roles and Responses. [4 hours]

Digital Forensics -Introduction to Digital Forensics - Forensic Software and Hardware - Analysis and Advanced Tools - Forensic Technology and Practices - Forensic Ballistics and Photography - Face, Iris and Fingerprint Recognition [6 hours]

Laws and Ethics - Digital Evidence Controls - Evidence Handling Procedures - Basics of Indian Evidence ACT, Legal Policies. [4 hours]

Outcomes:

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

* Understand the symmetric and asymmetric cryptographic algorithms.
* Describe common types of vulnerabilities and attacks in web applications, and defenses against them.
* Understand client side and server side security concepts and tools
* Propose and design security algorithm for a particular application
* Understand cybercrimes, cybercrime investigation, Laws and ethics.

References:

1. Mayank Bhushan, Fundamentals of cybersecurity, BPB publications, 2017
2. Raef Meeuwisse, Cyber Security for Beginners, 2015
3. Rolf Oppliger, Security Technologies for the World Wide Web, 2nd edition,Artech House, 2002.
4. Seth Fogie, Jeremiah Grossman, Robert Hansen and Anton Rager, XSS Attacks: Cross Site Scripting Exploits and Defense, Syngress, 2007.
5. Justin Clarke et.al.,SQL Injection Attacks and Defense, 2nd edition, Syngress, 2012.
6. DafyddStuttard, and Marcus Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws,2nd edition, Wiley, 2011.

|  |  |  |
| --- | --- | --- |
| **ICT 4307:** | **GAME THEORY WITH ENGINEERING APPLICATIONS** | **[3 0 0 3]** |

**Objectives:**

* To provide the comprehensive treatment of non-cooperative game theory
* To provide the comprehensive treatment of cooperative game theory
* To introduce the fundamental concepts of mechanism design
* To introduce game-theoretic approach to solve engineering problems

**Abstract:**

Introduction, Mathematical Preliminaries, Non-Cooperative Game Theory: Extensive Form Games, Strategies Form Games, Dominant Strategy Equilibria, Nash Equilibria, Matrix Games, Bayesian Games, Cooperative Game Theory: Two Person Bargaining Problem, Coalition Games, Shapely Values, Mechanism Design: Social Choice Functions, Incentive Compatibility and Revelation Theorem, Auctions

**Syllabus:**

**Introduction:**

Game theory-the science of strategic interactions, current trends and applications [**1 Hour**]

**Mathematical Preliminaries:**

Probability theory, Linear algebra, Linear programming and duality, Mathematical analysis, Computational complexity classes [**2 Hours**]

**Non-Cooperative Game Theory:**

Key notations in game theory, Extensive form games, Strategic form games, Dominant strategy equilibria, Pure strategy Nash equilibria, Mixed strategies and mixed strategy Nash equilibria, Matrix games, Bayesian games [**14 Hours**]

**Cooperative Game Theory:**

Correlated strategies and correlated equilibrium, Two person bargaining problem, Coalition games with transferable utility, Core of coalition games, Shapley values, Other solution concepts in cooperative game theory, Stable matching [**13 Hours**]

**Mechanism Design:**

Introduction to mechanism design, Social choice functions, Incentive compatibility and revelation theorem, Auctions [**6 Hours**]

**Learning Outcomes:**

On successful completion of this course, students will be able to:

* Identify strategic situations and represent them as games
* Solve simple games using various techniques
* Recommend and prescribe which strategies to implement
* Analyse engineering situations using game theoretic techniques

**References:**

1. Y Narahari, Game Theory and Mechanism Design, World Scientific, Chennai, 2015
2. Drew Fudenberg and Jean Tirole, Game Theory, ANE Books, New Delhi, 2015
3. Dario Bauso, Game Theoy with Engineering Applications, SIAM, Philadelphia, 2016