COMPUTER SCIENCE & ENGINEERING

Course Name	**:**	**INTRODUCTION TO COMPUTER SCIENCE AND ENGINEERING**
Course Code	**:**	**CSN101**
Credits	**:**	**2**
L T P	**:**	**2 0 0**

Course Objectives:
The students should know about various disciplines in Computer Science and Engineering and are aware of emerging trends of Computer Science and Engineering.

Total No. of Lectures – 28

Lecture wise breakup		**Number of Lectures**
1	**COMPUTER HARDWARE** Basics of Number System, Evolution of Computer Hardware, Moore's Law.	**4**
2	**LOGIC DEVELOPMENT AND ALGORITHMS** Various techniques to solve a problem, Ways to specify an algorithm, Flow charting techniques.	**4**
3	**VARIOUS DISCIPLINES OF COMPUTER SCIENCE AND ENGINEERING** Basics of Operating Systems, Artificial Intelligence, Computer Networks, Information Security, Software Engineering, Computer Vision.	**16**
4	Current and future trends and challenges in various fields of computing. Social, ethical and economical issues related to computing technology. Exploration of career and professional development opportunities.	**4**

Course Outcomes: At the end of the course, students will have:	
1	Knowledge about various fields of Computer Science and Engineering.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Computing Fundamentals, Peter Nortan, 4th Ed., Tata McRaw Hill	2001
2	Computer Science Handbook, Allen B. Tucker, CRC Press	2004

Course Name	**:**	**DATA STRUCTURES**
Course Code	**:**	**CSN 102**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
The students should be able to describe and implement various data structures including lists, arrays, stacks, queues, binary search trees, graphs, hash tables, and matrices. The student will be able to analyze and apply various algorithms for shortest path calculation, sorting and searching applications.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION** Introduction to Data Structures and data types, Efficient use of memory, Recursion, time and space complexity of algorithms, Big O Notation and theta notations.	**4**
2	**ELEMENTARY DATA STRUCTURES** Stacks, queues, Infix, Postfix & Prefix conversions, evaluations of expressions, multiple	**7**

	stacks and queues, priority queues as heaps, double ended queue, implementation of stacks and queues.	
3	**LINKED LISTS** Singly linked lists, linked stacks and queues, polynomial addition, sparse matrices, doubly linked lists and dynamic storage management, circular linked list, Applications of Stacks, Queues and Linked lists, Garbage collection, Josephus Problem.	**7**
4	**TREES** Basic terminology, binary trees, binary tree traversal, representations of binary tree, application of trees, decision tree, game trees, Threaded Trees, Binary Search Tree, AVL tree, B-tree.	**8**
5	**GRAPH THEORY** Graph representations, Graph Traversals, Dijkstra's algorithm for shortest path, Prim's and Kruskal's Algorithm for Minimal Spanning tree.	**8**
6	**SORTING AND SEARCHING** Searching: Linear search, binary search and hash search. Sorting: Insertion sort, selection sort, bubble sort, quick sort, merge sort, heap sort, and Bucket sort.	**8**

List of Experiments:		**Number of Turns**
1	Implement stack and its various applications.	**1**
2	Implement queue and its various applications.	**1**
3	Implement linked list and its various applications.	**2**
4	Implement binary trees and its various applications.	**2**
5	Implement AVL tree.	**1**
6	Implement binary search tree and its various applications.	**1**
7	Implement graphs.	**2**
8	Implement minimum spanning tree.	**2**
9	Implement various searching and sorting algorithms	**1**

Course Outcomes: At the end of the course, students will be able to:	
1	Choose the data structures that effectively model the information in a problem and analyze the efficiency trade-offs (run time and memory usage) among alternative data structure implementations or combinations.
2	Design, implement, test, and debug programs using a variety of data structures including stacks, queues, hash tables, binary and general tree structures, search trees, and graphs.
3	Use efficient data structure (linked lists, stacks and queues) to solve a particular problem.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Data Structures and Program Design in C By Robert L. Kruse,C.L. Tondo, Bruce Leung, Pearson Education.	2007
2	Data Structures Using C & C++, By Langsam, Augenstein, Tanenbaum, Pearson Education.	1989
3	Fundamentals of Data Structures, By Ellis Horowitz and Sartaj Sahni, Computer Science Press.	2011
4	An introduction to data structures with applications, By J.P. Trembley & P.G. Sorensen, TMH.	2004

Course Name	**:**	**DIGITAL ELECTRONIC AND LOGIC DESIGN**
Course Code	**:**	**CSN103**
Credits	**:**	**4**

L T P	:	3 0 2

Course Objectives:
Students should be able to apply the principles of Boolean algebra to manipulate and minimize logic expressions. Analyze the operation of sequential circuits built with various basic devices. Design combinational circuits using various devices.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION** Digital Systems; Data representation and coding; Logic circuits, integrated circuits; Analysis, design and implementation of digital systems.	**2**
2	**NUMBER SYSTEMS AND CODES** Positional number system; Binary, octal and hexadecimal number systems; Methods of base conversions; Binary, octal and hexadecimal arithmetic; Representation of signed numbers; Fixed and floating point numbers; Binary coded decimal codes; Gray codes; Error detection and correction codes - parity check codes and Hamming code.	**6**
3	**COMBINATIONAL LOGIC SYSTEMS** Definition and specification; Truth table; Basic logic operation and logic gates.	**4**
4	**BOOLEAN ALGEBRA AND SWITCHING FUNCTIONS** Basic postulates and fundamental theorems of Boolean algebra; Standard representation of logic functions - SOP and POS forms; Simplification of switching functions - K-map, Synthesis of combinational logic circuits.	**8**
5	**COMBINATIONAL LOGIC MODULES AND THEIR APPLICATIONS** Decoders, encoders, multiplexers, demultiplexers and their applications; Parity circuits and comparators; Arithmetic modules- adders, subtractors and ALU; Design examples.	**6**
6	**SEQUENTIAL LOGIC SYSTEMS** Definition of state machines, state machine as a sequential controller; Basic sequential circuits- latches and flip-flops: SR-latch, D-latch, D flip-flop, JK flip-flop, T flip-flop; Analysis of state machines using D flip-flops and JK flip-flops; Design of state machines - state table, state assignment, transition/excitation table, excitation maps and equations, logic realization; Design examples, Registers, counters, shift register, application examples.	**6**
7	**MEMORY** Introduction to Read-only memory, read/write memory - SRAM and DRAM	**4**
8	**LOGIC FAMILIES** Introduction to different logic families: RTL, TTL, Metal Oxide Semiconductor.	**2**
9	**PROGRAMMABLE LOGIC DEVICES** PLAs, PALs and their applications; Sequential PLDs and their applications; State-machine design with sequential PLDs; Introduction to field programmable gate arrays (FPGAs)	**4**

List of Experiments:		**Number of Turns**
1	Introduction and hands on with the simulation environment	**1**
2	Investigate the behavior of various logic gates (NAND, NOR, NOT, AND,OR, XOR).	**1**
3	Simulate a logic function using logic gates.	**1**
4	Design and simulate Adder and Subtractor circuits.	**1**
5	Design and simulate code converters.	**2**
6	Design and simulate Combinational circuits using Multiplexers.	**2**
7	Simulate Flip-flops using NAND and NOR Gates.	**1**
8	Simulate the operation of shift register.	**1**
9	Simulate the operation of counters.	**1**
10	Design and simulate the synchronous sequential circuits.	**1**
11	Design and simulate applications based on digital circuits.	**1**

Course Outcomes: At the end of the course, students will be able to:	
1	Design two-level logic functions with various gates.
2	Design combinational circuits using gates.
3	Design and build complex digital systems using state-of-the-art components.
4	Understand how to use state diagrams to design finite state machines using various types of flip-flops and combinational circuits.
5	Articulate how modern microelectronics has impacted society.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Digital Design by Morris Mano, 4th Ed. Pearson	2006
2	Digital logic and computer design by Morris Mano, Pearson Education Inc.	1979
3	Digital principles and applications by Malvino Leech, Tata McGraw Hill Education Pvt. Ltd	2006
4	Digital Electronics by R.P. Jain, Tata McGraw Hill Education Pvt. Ltd.	2003
5	Digital System Principals and Applications by R J Tocci, Pearson Education Inc.	2010

Course Name	**:**	**DISCRETE STRUCTURES FOR COMPUTER SCIENCE**
Course Code	**:**	**CSN 201**
Credits	**:**	**4**
L T P	**:**	**3 1 0**

Course Objectives:
Students should be able to understand Discrete Mathematical Structures (DMS) for the development of theoretical computer science, problem solving in Programming language using Discrete Structure and importance of discrete structures towards simulation of a problem in computer science and engineering.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**MATHEMATICAL REASONING** Mathematical reasoning, Propositions, Negation, disjunction and conjunction, Implication and Equivalence, Truth tables, Predicates, Quantifiers, Natural deduction, Rules of Inference, Methods of proofs, Resolution principle, Application to PROLOG.	**8**
2	**SET THEORY** Paradoxes in set theory, Inductive definition of sets and proof by induction, Peano postulates, Relations, Properties of relations, Equivalence Relations and partitions, Partial orderings, Posets, Linear and well-ordered sets.	**8**
3	**COMBINATORICS** Elementary combinatorics, Counting techniques, Recurrence relation, Generating functions	**4**
4	**FUNCTIONS** Functions; mappings, Injection and Surjections, Composition of functions, Inverse functions, Special functions, Pigeonhole principle, Recursive function theory	**6**
5	**GRAPH THEORY** Elements of graph theory, Euler graph, Hamiltonian path, trees, Tree traversals, Spanning trees, Representation of relations by graphs.	**8**
6	**GROUPS, RINGS, FIELDS** Definition and elementary properties of groups, Semigroups, Monoids, Rings, Fields, Vector spaces and lattices	**6**
7	**DISCRETE PROBABILITY** Introduction, Discrete random variables, Applications to Binary Search Tree.	**2**

Course Outcomes: At the end of the course, students will be able to:	
1	Acquire complete knowledge on various discrete structures.
2	Apply Applications of Discrete Structures in Computer Science and Engineering.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	K. H. Rosen, Discrete Mathematics and applications, 6th Edition, Tata McGraw Hill	2007
2	C. L. Liu, Elements of Discrete Mathematics, 2nd Edn., Tata McGraw-Hill	2000
3	J .L. Mott, A. Kandel, T.P .Baker, Discrete Mathematics for Computer Scientists and Mathematicians, Second edition, Prentice Hall of India	1986
4	W. K. Grassmann and J. P. Trembnlay, Logic and Discrete Mathematics, A Computer Science Perspective, Prentice Hall Inc	1996

Course Name	**:**	**COMPUTER ARCHITECTURE AND ORGANIZATION**
Course Code	**:**	**CSN202**
Credits	**:**	**4**
L T P	**:**	**3 1 0**

Course Objectives:
Students should be able to understand basic principles of Computer Systems. They should be able to understand various logic design techniques and their applications. They should be capable of analyzing the system performance.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**BASICS** Introduction to computers with block diagrams, Impact of technology. Designing combinational and sequential logic, computers registers and instructions, timing, and control, instructions cycle, memory reference instruction, I-O interruption.	**5**
2	**COMPUTER ARITHMETIC** Adder and Subtractor circuits, Booth Multiplication Algorithm, Performance bench marks.	**7**
3	**CONTROL PATH DESIGN** Sequence counter method, Micro programmed controllers address sequencing, symbolic microinstructions.	**7**
4	**CENTRAL PROCESSING UNIT** Registers General register organization, stack organization, instructions formats, address instructions, addressing modes, data transfer and manipulations, program control, RISC instruction set design, three address instructions and arithmetic pipelines with example of floating point adder, instructions pipelines , advanced pipelining using instruction level parallelism.	**10**
5	**MEMORY ORIGINATION** Memory device characteristics, Random Access Memory, Serial Access Memory, virtual memory, associative memory, cache memory, memory management hardware, hierarchy of various memories.	**7**
6	**I/O ORGANIZATION** I/O interface asynchronous data transfer, DMA interrupt, I/O processor.	**6**

Course Outcomes: At the end of the course, students will be able to:	
1	Design the organization of the Control unit, Arithmetic and Logical unit, Memory unit and the I/O unit.
2	Learn different computer architectures and hardware.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	M. Morris Mano, Computer System & Architecture, Prentice Hall of India	2002
2	John L. Hennessy and David A Patterson, Computer Architecture-A quantitative approach, Morgan Kaufmann/ Elsevier, 4th Edition	2007
3	Hayes .J.P, Computer architecture and organization by McGraw-Hill Companies	1998
4	M.Morris and Charles R. Kinre , Logic and computer design Fundamental, PHI	1995

Course Name	**:**	**OBJECT ORIENTED PROGRAMMING**
Course Code	**:**	**CSN203**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
The students should be able to understand the concept of object oriented programming like classes, constructors, polymorphism, inheritance, templates and file handling.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION TO PROGRAMMING PARADIGMS** Introduction to various programming paradigms, advantages of OOP, comparison of OOP with Procedural Paradigm	**2**
2	**C++ CONSTRUCTS** Tokens, Expressions and control structures, various data types and data structures, Variable declarations, Dynamic Initializations, Operators and Scope of Operators, Typecasting, Unformatted and formatted console I/O Operations	**2**
3	**FUNCTIONS, CLASSES AND OBJECTS** Prototyping, Referencing the variables in functions, Inline, static and friend functions. Memory allocation for classes and objects. Arrays of objects, pointers to member functions.	**4**
4	**CONSTRUCTORS AND DESTRUCTORS** Constructor and Destructor types, Dynamic Constructors, Applications, Order of Invocation, C++ garbage collection, dynamic memory allocation.	**5**
5	**POLYMORPHISM** Function and Operator overloading, overloading using friend Functions, type conversions from basic data types to user defined and vice versa.	**5**
6	**INHERITANCE** Base classes and Derived classes, types of inheritance, various types of classes, Invocation of Constructors and Destructors in Inheritance, aggregation, composition, classification hierarchies, metaclass/abstract classes.	**5**
7	**POINTERS** Constant pointers, Use of this Pointer, Pointer to derived and base classes, virtual functions, Bindings, Pure virtual Functions and polymorphism.	**5**
8	**I/O OPERATIONS AND FILES** Classes of files, Operations on file, file pointers.	**5**
9	**GENERIC PROGRAMMING WITH TEMPLATES** Definition of class template, Function Templates, Overloading Template Functions, Class templates and member functions templates with parameters, Standard C++ classes, persistent objects, streams and files, namespaces, exception handling, generic classes, standard	**4**

	template library: Library organization and containers, standard containers, algorithm and Function objects, iterators and allocators, strings, streams, manipulators, user defined manipulators and vectors.	
10	**ADVANCED MEMORY HANDLING** Storage types in C++, Automatic Life, Dynamic Life, Static Life, Object with special storage restrictions.	**2**
11	**CASE STUDY** Features of Different Object Oriented languages	**3**

List of Experiments:		**Number of Turns**
1	Implement various C++ constructs	**1**
2	Implement friend functions	**1**
3	Implement pointers to member functions	**2**
4	Implement constructors and destructors	**1**
5	Implement operator and function overloading	**2**
6	Implement inheritance	**1**
7	Implement run time polymorphism	**2**
8	Implement file operations	**2**
9	Implement templates	**1**

Course Outcomes: At the end of the course, students will be able to:	
1	Understand real world problem and identify object in given problem.
2	Construct C++ classes and apply various C++ concepts with proficiency

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	C++ Primer, 5th Edition, Pearson Education Inc.	2012
2	Object Oriented Programming in C++, Robert Lafore, 4th Edition, SAMS	2001
3	C++ Primer Plus By Prata, Pearson Education	2012
4	C++:The Complete Reference, By Schildt, McGraw-Hill	2003
5	Object Oriented Programming with C++, Balaguruswamy, Tata Mc Graw Hill	2008

Course Name	**:**	**ANALYSIS AND DESIGN OF ALGORITHMS**
Course Code	**:**	**CSN204**
Credits	**:**	**4**
L T P	**:**	**3 1 0**

Course Objectives:
The students should be able to analyze various algorithms mainly for time and space complexity. They should be able to develop algorithm for solving various computational problems by applying various algorithm design strategies. They should be able to understand the affect of choice of data structures on the complexity of algorithm.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**BASIC CONCEPTS OF ALGORITHMS** Introduction – Notion of Algorithm – Fundamentals of Algorithmic Solving – Important problem types – Fundamentals of the Analysis Framework – Asymptotic Notations and Basic Efficiency Classes.	**6**

2	**MATHEMATICAL ASPECTS AND ANALYSIS OF ALGORITHMS** Mathematical analysis of non-recursive algorithms. Mathematical analysis of recursive algorithm: recurrence relations, solution of recurrence relations using substitution method.	**5**
3	**BRUTE FORCE** Selection sort, Bubble sort, Sequential searching (Linear Search), Brute force string matching.	**4**
4	**DIVIDE AND CONQUER STRATEGY** General method, Merge sort, Quick Sort, Binary Search, Strassen's matrix multiplication	**4**
5	**GREEDY APPROACH** General method, Fractional Knapsack problem, Minimum cost spanning tree: Prim's and Kruskal's algorithm, Single source shortest path problem	**5**
6	**DYNAMIC PROGRAMMING** General method, Principle of optimality, Multi-stage graph problem, All pair shortest path problem, 0/1 Knapsack problem, Traveling salesperson problem.	**6**
7	**BACKTRACKING** General method, N-Queen problem, 0/1 Knapsack problem	**4**
8	**BRANCH AND BOUND** General method, 0/1 Knapsack problem, Traveling sales person problem	**4**
9	**LOWER BOUND THEORY AND COMPELXITY CLASSES** Lower bounds, Decision trees, P, NP and NP Complete problems	**4**

Course Outcomes: At the end of the course, students will be able to:	
1	Demonstrate the ability to select the best data structure for designing an algorithm to solve a given problem.
2	Demonstrate the ability to compare algorithms with respect to time and space complexity.
3	Develop algorithms to solve various computational problems.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Algorithm Design, Jon Kelinberg and Eva Tardos, 1st Edition, Pearson Education	2014
2	Fundamentals of algorithms, Horowitz E, Sahini S, Rajasekaran S., University Press	2008
3	Introduction to algorithms, Cormen, Leiserson, Rivest, Stein, 3rd Edition, PHI.	2012
4	An introduction to analysis of algorithms, R. Sedgewick, 1st edition, Pearson Education	1996
5	Data Structures and Program Design in C By Robert L. Kruse, C.L. Tondo, Bruce Leung, Pearson Education.	2007

Course Name	**:**	**ENGINEERING ANALYSIS AND DESIGN**
Course Code	**:**	**CSN206**
Credits	**:**	**4**
L T P	**:**	**3 1 0**

Course Objectives:
Students should be able to understand the concept and importance of Information Systems. They should be able to extract information requirements from any business system and design information systems by using Engineering Techniques used in industrial applications.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION TO INFORMATION SYSTEMS** Business Systems Concepts, Important systems characteristics, Business systems, Business	6

	Information Systems, Components of Information System, Categories of Information Systems, Scope of Information Systems, Modern Information Systems: Embedded, Mobile Applications, Big Data etc, Technology drivers for modern Information Systems	
2	**MANAGING THE APPLICATION DEVELOPMENT PORTFOLIO** Reasons for project proposals for information systems, Corporate missions and the role of information systems, Information System Building Blocks, Identifying and selecting Systems Development Projects, Initiating and Planning Systems Development Projects, Assessing Project Feasibility.	8
3	**REQUIREMENTS ANALYSIS AND DETERMINATION** Requirements determination, Basic Requirements, User Transaction Requirements, User Decision Requirements, Organization-wide Requirements, Fact-Finding Techniques, Tools for documenting procedures and decisions, Tools for Analysis and Determination, Structured Analysis.	8
4	**SYSTEMS MODELING AND DESIGN** Process Modeling, Data Modeling, Modeling Systems Requirements with Use Cases, System Design, Elements of Design, Logical design elements, Physical design elements, Features that must be designed, Transition from analysis to design, Systems design for In-House development-The Build solution, Systems design for Integrating Commercial Software-The Buy solution.	8
5	**APPLICATION ARCHITECTURE AND MODELING** Application Architecture, Physical Data Flow Diagrams, Information Technology Architecture, IT alternatives and trends influencing design decisions, Distributed Architecture, Centralized Systems, File Server Architecture, Client Server Architectures-Distributed Presentation, Distributed Data, Distributed Data and Application, Internet Based Computing Architectures, Data Architectures, Interface Architectures, Process Architectures, Modeling the Application Architecture of an Information System.	8
6	**FIELD CASE STUDIES** Various components of an Information Systems Project, Information Requirements of various Projects as Case studies Such as Various University Departments, Local City Bus Transport System, Local Radio-Taxi Booking System, Patient Registration System, Internet Based Computing-Online Shopping, Mobile Computing Applications etc.	4

Course Outcomes: At the end of the course, students will be able to:	
1	Understand the fundamentals of information system.
2	Understand about the various roles in information technology projects
3	Analyze, design and model various information system.

Suggested Books:

Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Analysis and Design of Information Systems by James A. Senn, McGraw-Hill, Computer Science Series.	1989
2	Systems Analysis and Design Methods by Whitten, Bentley and Dittman, Tata McGraw-Hill.	2001
3	Modern Systems Analysis and Design by Jeffrey A. Hoffer, Joey F. George and Joseph S. Valacich, Pearson Education Asia.	2008

Course Name	**:**	**MICROPROCESSOR AND ITS APPLICATIONS**
Course Code	**:**	**CSN207**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:

Students should be able to understand microprocessor based systems and interfacing techniques. Learn assembly language, assembly directives. Interface microprocessor with memories like RAM and ROM. Interface microprocessor with various peripheral devices. Overview of high end processors.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**ARCHITECTURE OF MICROPROCESSORS** Introduction to the general structure of 8 and 16 bit Microprocessors. Overview of 8085 microprocessor. Overview of 8086 microprocessor. Signals and pin diagram of 8086 microprocessor, Register structure, ALU, Bus Organization, Timing and Control.	**6**
2	**ARCHITECTURE OF A 16-BIT MICROPROCESSOR** Internal organization of 8086, Signal descriptions, Physical memory organization, BIU, EU, Minimum mode 8086 system and timings, Maximum mode 8086 system and timing.	**6**
3	**ASSEMBLY LANGUAGE OF 8086** Addressing modes, Instruction set, Assembler directives and Operators, Data movement instructions, Arithmetic and logic instructions, Program control instructions, Recursive procedures.	**10**
4	**SPECIAL ARCHITECTURAL FEATURES AND RELATED PROGRAMMING** Stack structure, Interrupts and Interrupt service routine, Interrupt programming, Macros, Timings and delays.	**5**
5	**BASIC PERIPHERALS AND THEIR INTERFACING** Memory interfacing, Interfacing I/O ports, Programmable Peripheral Interface (8255), Interfacing A/D and D/A converters.	**3**
6	**SPECIAL PURPOSE PROGRAMMABLE PERIPHERAL DEVICES AND THEIR INTERFACING** Programmable Interval Timer (8253/8254), Programmable Interrupt Controller (8259), Keyboard/Display Controller (8279), Programmable Communication Interface (8251), DMA Controller (8237/8257).	**5**
7	**MICROPROCESSOR'S APPLICATION** Interfacing scanned multiplexed displays and Liquid crystal displays, Interfacing matrix keyboard, Stepper motor interfacing, Case studies of microprocessor based systems, Standards for bus architecture and ports.	**5**
8	**HIGH END PROCESSORS** Introduction to Pentium and onwards	**2**

List of Experiments:		**Number of Turns**
1	To introduce Assembly Language, Assembly Language Fundamentals	1
2	Assembling, Linking and Debugging, Data Allocation Directives	1
3	Programming Data Transfer Instructions (8086 Instruction Set)	2
4	Programming Arithmetic Instructions (8086 Instruction Set)	2
5	Programming Multiply and Divide Instructions (8086 Instruction Set)	1
6	Programming Logical Instruction (8086 Instruction Set)	1
7	Programming Shift Rotate Instructions (8086 Instruction Set)	1
8	Programming Transfer Control Instructions "CALL and RET" (8086 Instruction Set)	1
9	Programming Transfer Control Instructions "Jump" (8086 Instruction Set)	1
10	Programming Transfer Control Instructions "Conditional Jump" (8086 Instruction Set))	1
11	Programming Stack Operation Instructions "PUSH & POP" (8086 Instruction Set)	1

Course Outcomes: At the end of the course, students will be able to:	
1	Program a microprocessor based system.
2	Interface a microprocessor with various peripheral devices.
3	Implement real time embedded applications
4	Use microprocessor peripherals effectively

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	The Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386, Pentium, and Pentium Pro Processor- B.B. Brey, Pearson Education	2009
2	Advanced Microprocessors and Peripherals, Ray and Bhurchandi, 2nd Edition, Tata McGraw Hill	2006
3	Microprocessor and Interfacing - D. V. Hall, McGraw Hill Publishing Co.	1990
4	Microprocessors and Microcomputer based System Design- Rafiquzzaman , PHI	2002

Course Name	**:**	**DATA BASE MANAGEMENT SYSTEM**
Course Code	**:**	**CSN208**
Credits	**:**	**3 0 2**
L T P	**:**	**4**

Course Objectives:
Students should be able to understand various applications of DBMS and query languages.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION** Introduction and application of DBMS, Data Independence, Database System Architecture – levels, Mapping, Database users and DBA, Database Languages: DDL, DML **STORAGE AND FILE STRUCTURE** Overview of physical storage media, magnetic disks, RAID, file organization, organization of records in files, indexing and hashing **DATABASE MODELS:** Entity – Relationship model, constraints, keys, Design issues, E-R Diagram, Weak and Strong entity types, Extended E-R features- Generalization, Specialization, Aggregation, Translating E-R model into Relational model Network model, Hierarchical model	8
2	**RELATIONAL MODE**L: Introduction to relational model, basic structure, Types, Keys, views in a relational database. **SQL:** Fundamentals, basic structure, set operations, aggregate operations, DDL, DML, DCL, nested queries, complex queries, Integrity Constraints, PL/SQL Concepts, triggers	8
3	**RELATIONAL ALGEBRA AND RELATIONAL CALCULUS** Relational Algebra: Fundamental operations, Additional Operations Relational calculus: Tuple Relational calculus, Domain Relational calculus	5
4	**RELATIONAL DATABASE DESIGN:** Functional Dependencies, Non-loss Decomposition, First, Second, Third Normal Forms, Dependency Preservation, Boyce/Codd Normal Form, Multi-valued Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form.	9
5	**TRANSACTION MANAGEMENT** Transaction concept, transaction state, ACID properties, serializability, Recoverability , Implementation of Isolation , Testing for serializability . concurrency control, Lock based concurrency control, Time stamping methods	7
6	**DISTRIBUTED DATABASES** Introduction, data mining, Big Data, No SQL, New SQL, Modern databases based on these concepts.	5

List of Experiments:		Number of Turns
1	For a given scenario of database application, Create the required tables using SQL Commands	1
2	Write Sql queries to apply the constraints i.e. Primary Key, Foreign key, UNIQUE to the tables.	2
3	SQL queries for Null values and different clauses	2
4	Usage of SELECT, rename, tuple operations, DELETE etc.	2
5	SQL queries for implementing various String operations, Set operations	2
6	SQL queries for implementing JOINs and types of joins with conditions.	1
7	SQL nested queries for a particular scenario.	1
8	SQL queries to create the views ,triggers	1
9	SQL queries to create indexes and apply on a database.	1

Course Outcomes: At the end of the course, students will be able to:	
1	Design a database in various environments
2	Design , implement, test and debug the SQL queries
3	Design a database with triggers for maintaining the consistency of the database

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	"Database System Concepts", Abraham Silberschatz, Henry F. Korth, S. Sudharshan, Tata McGraw Hill,	2006
2	"Fundamentals of Database Systems", Elmsari and Navathe, Pearson Education	2013
3	Database Management Systems , Ramakrishnan and Gehrke, McGrawHill	2003
4	"An Introduction to Database Systems", C.J.Date, A.Kannan, S.Swamynathan, Pearson Education	2006
5	J. D. Ullman, "Principles of Database Systems", 2nd Ed., Galgotia Publications	1999

Course Name	**:**	**OPERATING SYSTEMS**
Course Code	**:**	**CSN209**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
Students should be able to describe the purpose, structure and functions of operating system. They should be able to understand the file system, processes synchronization, process scheduling, system calls and memory management techniques used in an operating system.

Total No. of Lectures – 42

Lecture wise breakup		Number of Lectures
1	**CONCEPTS OF OPERATING SYSTEMS** Computer system overview, concept of an operating system, batch system, multiprogramming, multiprocessing, multi user, time sharing, personal system, parallel system, real time system, simple monitors, general system architecture, System components, operating system services, system calls, system programs, system structure, Approaches to OS design and implementation: Microkernel, Layered, Kernel Approach	**8**
2	**PROCESSES AND THREADS** Concept of process, process states, process state transitions, process control block, operations on processes, threads, concurrent processes, mutual exclusion and	**8**

	synchronization, principles of deadlocks, integrated deadlocks strategy, scheduling levels, scheduling criteria, Inter process synchronization, Inter process communication, Linux, IPC Mechanism, Remote procedure calls, RPC exception handling, security issues	
3	**MEMORY MANAGEMENT** Logical and physical address space, storage allocation and management techniques, swapping concepts of multi programming, paging, segmentation, virtual storage management strategies, demand paging, page replacement algorithm, thrashing	**8**
4	**INPUT/OUTPUT AND DATA MANAGEMENT** File organization, record blocking, access method, directory structure, protection file system structure, allocation methods, free space management, directory implementation, disk structure, disk scheduling, disk management, buffering, swap space management, RAID levels	**8**
5	**OS SECURITY** Types of Threats in OS, Basic OS Security Mechanisms, Understanding the Threats: Malware Taxonomy: Viruses, Worms, Rootkits, Defense -- An Overview, Logging, Auditing, and Recovery, OS-level Memory Protection	**4**
6	**CASE STUDIES** Linux/Unix OS design and architecture, Unix shell, Unix operating system services, user perspective, representation of files in Unix system processes and their structure, input-output system, memory management in Unix	**4**
7	**OS ABSTRACTIONS** Processes: fork, wait, exec, exit, kill, getpid, brk, nice, sleep, trace Files: open, close, read, write, lseek, stat, sync Directories: mkdir, rmdir, link, unlink, mount, umount users + Security: chown, chmod, getuid, setuid, Inter process communication: signals, pipe Networking: socket, accept, snd, recv, connect	**2**

List of Experiments:		**Number of Turns**
1	To perform shell programming	2
2	Implement memory management techniques like paging or segmentation	2
3	Implement any file allocation technique (Linked, Indexed or Contiguous)	2
4	Use the system calls of UNIX operating system: mkdir, rmdir, link, unlink, mount, umount users +	2
5	Use the following system calls of UNIX operating system: fork, wait, exec, exit, kill, getpid, brk, nice, sleep, trace, open, close, read, write, lseek, stat, sync	2
6	Use the following system calls of UNIX operating system: mkdir, rmdir, link, unlink, mount, umount users +, chown, chmod, getuid, setuid	2
7	Use the following system calls of UNIX operating system: signals, pipe, socket, accept, snd, recv, connect	1

Course Outcomes: At the end of the course, students will be able to:	
1	Explain about operating systems and its major components.
2	Write and/or modify concurrent programs.
3	Apply security as well as recovery feature in the design of algorithms.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Operating system, Galvin & Silberschatz, 7th Edition, John Willey	2004
2	Operating Systems-A Concept Based Approach By Dhamdhare, TMH	2006
3	Operating systems Internals and design principles By William Stallings, Pearson Education	2012
4	Operating Systems –A Design Oriented approach By Crowley, TMH	2001

5	Operating systems Design and Implementation By Andrew S. Tanenbaum, Pearson Education	2009

Course Name	**:**	**COMPUTER NETWORKS**
Course Code	**:**	**CSN210**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:

Students should be able to have an understanding of the fundamental concepts of computer networking and have a basic knowledge of the various networks models and their uses. They should be able to understand the organization of computer networks, factors influencing computer network development and the reasons for having variety of different types of networks. They should be able to analyze simple protocols and independently study literature concerning computer networks.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**COMPUTER NETWORKS AND THE INTERNET** What is the Internet; network edge; network core; Delay, Loss and throughput in Packet-Switched Networks; Protocol Layers and their Service Models.	6
2	**APPLICATION LAYER** Principles of Network Applications; The Web and HTTP; File Transfer: FTP;Electronic Mail in the Internet; DNS - The Internet's Directory Service; Peer-to-Peer applications; Socket Programming – Creating network applications.	8
3	**TRANSPORT LAYER** Introduction and Transport-Layer Services; Multiplexing and Demultiplexing; Connectionless Transport: UDP; Principles of Reliable of Data Transfer; Connection-Oriented Transport: TCP; Principles of Congestion Control;TCP Congestion Control.	10
4	**NETWORK LAYER** Introduction; Virtual circuit and datagram networks; What is inside a router; Internet Protocol (IP): Forwarding and Addressing in the Internet; Routing Algorithms; Routing in the Internet; Broadcast and Multicast Routing.	10
5	**DATA LINK LAYER** Introduction to the link layer; Error Detection and Correction Techniques; Multiple Access links and Protocols; Switched local area networks.	8

List of Experiments:		**Number of Turns**
1	Programs using TCP and UDP Sockets	2
2	Simulation of sliding window protocols	2
3	Simulation of Routing Protocols	3
4	Configure given network topologies using any network simulator software	3
5	Programs for error detecting codes, RSA algorithm	2
6	Programs for Client server Communication	1

Course Outcomes: At the end of the course, students will be able to:	
1	Independently understand basic computer network technology.
2	Understand and explain various components of computer networks.
3	Identify the different types of network topologies and protocols.
4	Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.
5	Identify the different types of network devices and their functions within a network.
6	Understand and build the skills of routing mechanisms.

7	Familiarize with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	James F. Kurose and Keith W. Ross, "Computer Networking: A top down approach", Pearson Education, 6th edition.	2012
2	A.S. Tanenbaum, "Computer Networks", 5th Edition, PHI	2010
3	G. Keiser, "Local Area Networks", 2nd Edition, TMH	2002
4	D. Bertesekas and R. Gallager, "Data Networks", 2nd Edition, PHI	2000
5	William Stallings, "Data & Computer Communication", PHI, 10th Edition	2013
6	B.A. Forouzan, "Data communications and networking", TMH, 5th Edition	2012
7	B.A. Forouzan, "Local Area Networks", TMH.	2002
8	B.A. Forouzan, "TCP/IP Protocol Suite", TMH.	2004

Course Name	**:**	**THEORY OF COMPUTATION**
Course Code	**:**	**CSN301**
Credits	**:**	**4**
L T P	**:**	**3 1 0**

Course Objectives:
Students should be able to understand fundamental mathematical and computational principles that are foundations of computer science. They should learn about abstract models of computation, finite representations for languages and will gain formal understanding of algorithms and procedures.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**AUTOMATA** Introduction to formal proof, Additional forms of proof, Inductive proofs, Finite Automata (FA), Deterministic Finite Automata (DFA), Non-deterministic Finite Automata (NFA), Finite Automata with Epsilon transitions.	**9**
2	**REGULAR EXPRESSIONS AND LANGUAGES** Regular Expression, FA and Regular Expressions, Proving languages not to be regular, Closure properties of regular languages, Equivalence and minimization of Automata	**9**
3	**CONTEXT-FREE GRAMMARS AND LANGUAGES** Context-Free Grammar (CFG), Parse Trees , Ambiguity in grammars and languages, Definition of the Pushdown automata, Languages of a Pushdown Automata, Equivalence of Pushdown automata and CFG Deterministic Pushdown Automata.	**9**
4	**PROPERTIES OF CONTEXT-FREE LANGUAGES** Normal forms for CFG, Pumping Lemma for CFL, Closure Properties of CFL, Turing Machines, Programming Techniques for TM, Variations of TM, Non-Universal TM, Universal TM.	**8**
5	**UNDECIDABALITY** A language that is not Recursively Enumerable (RE), An undecidable problem that is RE Undecidable problems about Turing Machine, Post's Correspondence Problem, The classes P and NP.	**7**

Course Outcomes: At the end of the course, students will be able to:	
1	Express computer science problems as mathematical statements and to formulate proofs.

2	Become proficient in key topics of theory of computation, and to have the opportunity to explore the current topics in this area.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	J.E. Hopcroft, R. Motwani and J.D. Ullman, "Introduction to Automata Theory, Languages and Computations", second Edition, Pearson Education	2007
2	H.R. Lewis and C.H. Papadimitriou, "Elements of the theory of Computation", Second Edition, Pearson Education	2003
3	Thomas A. Sudkamp," An Introduction to the Theory of Computer Science, Languages and Machines", Third Edition, Pearson Education.	2007
4	J. Martin, "Introduction to Languages and the Theory of computation" Third Edition, Tata Mc Graw Hill.	2007

Course Name	**:**	**SOFTWARE ENGINEERING**
Course Code	**:**	**CSN302**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
Students should learn the concept and importance of Software Engineering. They should be able to construct software that is reasonably easy to understand, modify, maintain and reliable. They should learn strengths and weaknesses of various Software Engineering Techniques used in industrial applications.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION TO SOFTWARE ENGINEERING** Software, Software Engineering, Myths, Software Process, Work Products, Importance of Software Engineering.	**4**
2	**SOFTWARE PROCESS MODELS** Standard for Software Process, Waterfall Model, Prototyping Model, Iterative Enhancement Model, Spiral Model, RAD model, 4th Generation models, Formal Methods, Agile Development.	**6**
3	**REQUIREMENT ENGINEERING** Software Requirements, Types of Requirements, Requirement Engineering Cycle, Requirements Specification document, Characteristics of Requirements, Requirement verification and validation.	**4**
4	**SOFTWARE PROJECT MANAGEMENT** Role of Management in Software Development, Project Estimation Techniques, Staffing, Scheduling, Earned Value Analysis, Software Risks, Software Configuration Management, Software Process and Project metrics.	**6**
5	**SOFTWARE DESIGN** Process, Data and Behavioral Modeling, Design Concepts, Modularity, Architectural design, Coupling and Cohesion, Top-down and bottom-up design, Object-oriented Analysis, Function-oriented and Object-Oriented Design approach, Software Design Document.	**6**
6	**CODING AND TESTING** Coding styles and documentation, Testing principles, Testing strategies, Black-box and White-box Testing Techniques, Levels of testing -unit, integration, system, regression, Test Plan, Test Cases Specification, Software debugging, Software Maintenance.	**6**
7	**SOFTWARE QUALITY**	**4**

	Software Quality Assurance (SQA), SQA tasks, Software amplification and removal, Formal Technical Reviews, Software Quality Factors, ISO 9126, SEI CMM, CMMI, Software Reliability. Software Availability.	
8	**COMPUTER AIDED SOFTWARE ENGINEERING** Computer Aided Software Engineering (CASE) and its Scope, CASE support in Software Life Cycle, Architecture of CASE Environment, Upper CASE and Lower CASE, Exposure to CASE tools.	**3**
9	**ADVANCED TOPICS IN SOFTWARE ENGINEERING** Software Process Improvement, Component Based Software Engineering, Web Engineering, Reverse Engineering, Software Engineering challenges of Big Data, Mobile Applications.	**3**

List of Experiments:		**Number of Turns**
1	Project Development with Software Engineering practices.	3
2	Programming Exercises for software design concepts.	3
3	Programming Exercises for software testing concepts.	3
4	Programming Exercises for Project Management concepts.	2
5	Exposure to UML, Rational software Architect.	2

Course Outcomes: At the end of the course, students will be able to:	
1	Design software in various application domains
2	Design software solution independently as well as in teams

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Software Engineering-A Practitioners Approach, By R. Pressman, McGraw Hill International edition.	2004
2	Software Engineering, Ian Sommerville, Addison-Wesley.	2010
3	An Integrated Approach to Software Engineering, Pankaj Jalote, Narosa.	2014
4	Fundamentals of Software Engineering, By Rajib Mall, PHI Learning Pvt. Ltd.	2014
5	Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, New Age International Publishers	2007

Course Name	**:**	**WEB TECHNOLOGIES**
Course Code	**:**	**CSN303**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
Students should understand fundamentals of programming in Java. They should be able to design and develop Java programs using JDK environment. They should learn the basics about Client side scripts and Server side scripts.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION TO JAVA** Creation of Java, importance to Internet, Java applets and applications, security, portability, Java's Byte code. Understanding the java programs, compiling the program, control statements, lexical issues, Data types in Java, Java literals, variables, scope and lifetime of variables, type conversion,	4

	declaring and using arrays	
2	**OPERATORS IN JAVA** Arithmetic, Modulus, Assignment, Bitwise, Relational, Assignment, ternary operator, Operator precedence. Control statements, classes, garbage Collection, overloading methods, overloading constructors, using objects as parameters. Argument passing by value and by reference, returning objects, static and final keywords. Inheritance in Java using super keyword, abstract classes.	6
3	**EXCEPTION HANDLING IN JAVA** Exception types, using try and catch, multiple catch classes, nested try statement, using throw, Java Built in Exceptions, creating user defined exceptions.	6
4	**MULTITHREADED PROGRAMMING IN JAVA** Java thread model, Thread priorities, synchronization, creating thread using thread class and runnable interface, creating multiple threads, deadlocks.	6
5	**APPLETS AND AWT** Applet fundamentals, Applet architecture, creating and running applets, Passing parameters in Applets. AWT : AWT Basics, AWT classes, Working with event frame windows, working with graphics, creating and selecting a font, Managing Text output using Font Metrics.	6
6	**USING AWT CONTROLS** Menus, using Labels, buttons, checkboxes, Checkbox group, choice controls, using Lists, managing scrollbars, using a textField, textArea. Layout manager FlowLayout, borderLayout, gridLayout, menubars, menus.	4
7	**SCRIPTING** Client side scripting : Java script Server side scripting: Java server pages	6
8	**CONTENT MANAGEMENT SYSTEM** HTML5,CSS, Wordpress, druppal, Joomla	4

List of Experiments:		**Number of Turns**
1	Java program based on operators, loop, decision making statement and type casting.	1
2	Java program to define class, constructors and method overloading.	1
3	Java programs based on different types of inheritance.	1
4	Java program to demonstrate the use of interfaces.	1
5	Java program to demonstrate the use of packages.	1
6	Java programs based on exception handling.	1
7	Java program based on multithreading and synchronization.	1
8	Program to design simple applet using graphics class.	1
9	Program to design simple applet using AWT class.	1
10	Web based project using Applet, AWT class, client side scripting and server side scripting concepts.	4

Course Outcomes: At the end of the course, students will be able to:	
1	Demonstrate object-oriented design principles including encapsulation and information hiding, inheritance.
2	Write, compile and execute Java programs to solve problems.
3	Design and develop projects using concepts like Applet, AWT class, client and server side scripting.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	The Complete Reference Java 2, 5th Edition, by Herbert Schildt, McGraw-Hill	2007
2	Programming with JAVA, by Balagurusamy, Tata McGraw-Hill publishing company limited	2009

3	Professional wordpress: design and development, 3rd Edition, by Hal Stern, David Damstra, Brad Williams, Wiley India Private Limited	2015
4	JOOMLA BIBLE,1st Edition, by RIC SHREVES, Wiley	2010
5	BEGINNING DRUPAL, 1st Edition, by Jacob Redding, Wiley	2010
6	JavaScript the complete reference, 3rd Edition, by Thomas A Powell, Fritz Schneiider, McGraw-Hill	2012

Course Name	**:**	**COMPUTER GRAPHICS**
Course Code	**:**	**CSN304**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:

The students should be able to understand contemporary terminology, progress, issues and trends in the area of computer graphics. They should be able to design algorithms for drawing various graphical entities like lines, circles, ellipses and other graphical objects. They should be able to transform various graphical objects in 2-D and 3-D.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**GRAPHICS HARDWARE** Introduction, Application areas of Computer Graphics, overview of graphics systems, video-display devices, and raster-scan systems, random scan systems, graphics output and input devices.	**4**
2	**OUTPUT AND FILLED AREA PRIMITIVES** Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms, Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms.	**6**
3	**2-D GEOMETRIC TRANSFORMATIONS** Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems	**6**
4	**2-D VIEWING** The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Liang-Barsky line clipping algorithms, Sutherland –Hodgeman and Weiler-Atherton polygon clipping algorithm.	**6**
5	**3-D GEOMETRIC TRANSFORMATIONS** Translation, rotation, scaling, reflection and shear transformations, composite transformations. 3-D projections.	**6**
6	**3-D OBJECT REPRESENTATION** Polygon surfaces, quadric surfaces, spline representation, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces. Basic illumination models, polygon-rendering methods.	**4**
7	**VISIBLE SURFACE DETECTION** Classification, back-face detection, Hidden surface removal algorithms.	**4**
8	**ILLUMINATION MODELS AND SHADING** Gouraud Shading, Phong Shading.	**3**
9	**INTERACTIVE COMPUTER GRAPHIC TECHNIQUES** Inking, Trailing, Rubber-band techniques, dumb-bell shape of line	**3**

List of Experiments:		**Number of Turns**
1	Implement various line drawing algorithms such as DDA, Bresenham, Mid-point line drawing algorithms, etc.	**3**

2	Implement various circle drawing algorithm.	**2**
3	Implement ellipse drawing algorithm.	**1**
4	Implement 2-D transformation.	**4**
5	Implement 3-D projections and 3-D transformations.	**3**

Course Outcomes: At the end of the course, students will be able to:	
1	Understanding of contemporary graphics hardware.
2	Design interactive graphics applications using C, C++ and other libraries like OPENGL
3	Understanding of object hierarchy in graphics applications.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Computer Graphics with OPENGL, Donald Hearn and M. Pauline Baker, Pearson Education	2014
2	Computer Graphics: Principles and Practices, J. D. Foley, A. Van Dam, S. K. Feiner and R. L. Phillips, Addison Wesley	1996
3	Computer Graphics, F S Hill Jr, Pearson Education	2003
4	OpenGL programming Guide, Shreine, Woo, Neider and Davis, Pearson Education	2008

Course Name	**:**	**ARTIFICIAL INTELLIGENCE**
Course Code	**:**	**CSN305**
Credits	**:**	**4**
L T P	**:**	**3 1 0**

Course Objectives:
Students should learn the basic concepts and techniques of Artificial Intelligence. They should be able to develop AI algorithms for solving practical problems.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION:** Artificial Intelligence and its applications, Artificial Intelligence Techniques, Level of models, criteria of success, Intelligent Agents, Nature of Agents, Learning Agents. AI Techniques, Importance, functions, advantages, and limitations of AI	4
2	**PROBLEM SOLVING TECHNIQUES:** State space search, control strategies, heuristic search, problem characteristics, production system characteristics., Generate and test, Hill climbing, best first search, A* search, Constraint satisfaction problem, Mean-end analysis, Min-Max Search, Alpha-Beta Pruning, Additional refinements, Iterative Deepening	8
3	**KNOWLEDGE REPRESENTATION SCHEMES:** Mapping between facts and representations, Approaches to knowledge representation,	6
4	**LOGIC:** Propositional logic, predicate logic, Resolution, Resolution in proportional logic and predicate logic, Clause form, unification algorithm,	4
5	**KNOWLEDGE REPRESENTATION AND REASONING:** procedural vs declarative knowledge, Forward vs. Backward reasoning, Matching, conflict resolution, Non-monotonic reasoning, Default reasoning, statistical reasoning, fuzzy logic Weak and Strong filler structures, semantic nets, frame, conceptual dependency, scripts	6
6	**PLANNING:** The Planning problem, planning with state space search, partial order planning, planning graphs, planning with propositional logic, Analysis of planning approaches, Hierarchical planning, conditional planning, Continuous and Multi Agent planning	6

7	**NATURAL LANGUAGE PROCESSING AND EXPERT SYSTEM:** Basic Tasks of Natural Language processing, Expert systems, Expert system examples, Expert System Architectures, Rule base Expert systems, Non Monotonic Expert Systems, Decision tree base Expert Systems.	5
8	**AI PROBLEMS:** Pattern (biological sequence) recognition, Voice recognition, Feature extraction	3

Course Outcomes: At the end of the course, students will be able to:	
1	Solve problems by applying suitable AI techniques
2	Apply knowledge representation schemes for designing knowledgebase
3	Solve the challenges in the field of AI

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Artificial Intelligence: A modern approach by Stuart Russel, Pearson Education	2010
2	Artificial Intelligence by Rich and Knight, TMH	2003
3	Artificial Intelligence: A new synthesis by Nils and Nilson, Elsevier	1997
4	Artificial Intelligence by Luger, Pearson Education	2008
5	Artificial Intelligence by Padhy, Oxford Press	2005
6	Introduction to Artificial Intelligence by Charniak and Mcdermott, Addison-Wesley	1985

Course Name	**:**	**COMPILER DESIGN**
Course Code	**:**	**CSN401**
Credits	**:**	**4**
L T P	**:**	**3 1 0**

Course Objectives:
The students should be able to understand the concept of language translation and compiler design. They should learn various parsing techniques. They should develop practical programming skills for constructing a compiler.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**COMPILER STRUCTURE** Analysis-synthesis model of compilation, various phases of a compiler, passes of compilers, bootstrapping, tool based approach to compiler construction.	**4**
2	**PHASES OF COMPILERS** Lexical analysis: Interface with input, parser and symbol table, token. Difficulties in lexical analysis. Error reporting. Implementation. Regular definition, One-pass compilation techniques, Transition diagrams, implementation techniques, use of lexical analyzer generators LEX, specific source language issues.	**4**
3	**SYNTAX ANALYSIS AND BASIC PARSING TECHNIQUES** Syntax directed definitions like Inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top down evaluation of attributes, L- and S-attributed definitions. CFGs, derivations and parse trees, ambiguity, associativity, precedence, use of syntax analyzer generators, top down parsing, shift reduce parsing, recursive descent parsing, transformation on the grammars, predictive parsing, bottom up parsing, operator precedence grammars, LR(k) parsing (SLR, LALR, LR), YACC.	**6**
4	**AUTOMATIC CONSTRUCTION OF SOME EFFICIENT PARSERS**	**6**

	Canonical collection of LR(0) items, constructing SLR parsing tables, constructing LR parsing table, constructing LALR parsing tables, ambiguous grammars usages, implementation of LR parsing tables, constructing LALR sets of items.	
5	**INTERMEDIATE CODE GENERATION** Syntax directed translation schemes and their implementation, Intermediate languages, quadruples and triples, assignment statements, boolean expressions, array references, procedure calls, declarations, case statements.	**4**
6	**SEMANTIC ANALYSIS** Type checking, type system, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions.	**4**
7	**RUN TIME SYSTEM AND OPTIMIZATION** Storage organization, activation tree, activation record, parameter passing, symbol tables, data structures for symbol tables, handling recursive calls, global optimization through flow graph analysis, dynamic storage allocation, local optimization techniques, loop optimization techniques, loop-invariant, peephole optimization.	**4**
8	**ERROR DETECTION AND RECOVERY** Introduction to errors in all phases of compilers, lexical-phase errors, synthetic phase errors, semantic errors and various recovery methods.	**5**
9	**CODE GENERATION AND INSTRUCTION SELECTION** Basic blocks and flow graphs, register allocation, code generation, DAG representation of programs, code generation from dags, code generator generators, specifications of machine, Compiler-Compilers, Parser generators, machine independent code generation	**5**

Course Outcomes: At the end of the course, students will be able to:	
1	Use of lex tool & yacc tool to develop a scanner & parser.
2	Design experiments for Intermediate Code Generation and translation in compiler.
3	Design & implement a software system for backend of the compiler.
4	Code optimization in terms of speed & space.

Suggested Books:

Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	A. V. Aho, J. D. Ullman, M. S. Lam, R. Sethi. Compilers: Principles, Techniques and Tools , Addison-Wesley.	2014
2	C. Fischer and R. LeBlanc. Crafting a Compiler , Benjamin Cummings	2007
3	A. C. Holub. Compiler Design in C , Prentice-Hall Inc.	2002
4	K. Louden, Compiler Construction, Principles and Practice, Cengage Learning	1997

Course Name	**:**	**SOFTWARE TESTING**
Course Code	**:**	**CSN402**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:

Students should be able to learn fundamentals of software Development life cycle and identify relationship between testing and development. They should be able to identify the testing challenges associated with several applications.

Total No. of Lectures – 42

Lecture wise breakup	**Number of Lectures**

1	**FUNDAMENTALS OF TESTING** Errors, Faults, Failures and Defects in software, Fundamentals of Test Process, General Principles of Testing, Test Metrics, Role of Testing in SDLC, Comparing Software and Hardware Testing, Verification and Validation, Exhaustive testing, Impractical complete testing, All Test paths.	**6**
2	**SOFTWARE TESTING APPROACHES** Introduction to Test cases, test case design, Levels of testing - module, integration, system, regression, Structural versus Functional Technique Categories, Static versus Dynamic Testing, Control flow & Data flow testing, Determining Metrics, Black Box Testing, White Box Testing.	**10**
3	**TEST MANAGEMENT** Test Organization, Test Planning, Test Strategies, Test Prioritization, Performance, Load, Stress & Security Testing, Debugging.	**4**
4	**OBJECT ORIENTED TESTING** Object Oriented Testing Issues, OO Testing Methodologies, Analysis and Design Testing-UML Based, Class Testing, Integration Testing, Testing Hierarchies.	**8**
5	**DIFFERENT SOFTWARE TESTING TECHNIQUES** GUI testing, Validation testing, Regression testing, Scenario testing, Specification based testing, Adhoc testing, Smoke testing, Random Testing, Availability and Safety, Advances in Software Testing Methods.	**8**
6	**TEST AUTOMATION** Software test automation, scope of automation, design and architecture for automation, requirements for a test tool, challenges in automation, Test metrics and measurements, Types of reviews.	**6**

List of Experiments:		**Number of Turns**
1	Exercise for Functional test/ Black box Testing.	**2**
2	Programming Exercises for Dynamic memory by using graphs.	**2**
3	Programming Exercise for Control flow graph, independent paths, cyclomatic complexity and generation of test cases.	**2**
4	Programming Exercise for Data flow testing	**1**
5	Combination of paths and flow analysis	**1**
6	Exposure to Automated Testing Tools & Case studies, Study of Testing tools	**2**
7	Minor project	**3**

Course Outcomes: At the end of the course, students will be able to:	
1	Generate test cases effectively from Requirements, Design Models, Code etc.
2	Generate test cases automatically.
3	Apply various test cases to industrial applications.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Boris Beizer, Software Testing Techniques, John Wiley & Dreamtech.	2003
2	Aditya P. Mathur, Foundations of Software Testing, Pearson Education.	2008
3	Yogesh Singh, Software Testing, Cambridge University Press.	2012
4	William E. Perry, Effective Methods for Software Testing, John Wiley & Sons.	2006
5	Glenford J. Myers, The Art of Software Testing, Wiley India Pvt. Ltd.	2006
6	John D. McGregor & David A, A practical guide to testing object-oriented software, Addison-Wesley object technology series.	2001

Course Name	**:**	**SOFT COMPUTING**
Course Code	**:**	**CSN403**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
Students should be able to understand soft computing concepts and techniques and foster their abilities in designing and implementing soft computing based solutions for real-world problems.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION TO NEURAL NETWORKS** Structure and working of Biological Neural Network, Fundamentals of Artificial Neural Networks & Applications, Characteristics of Artificial Neural Networks, History of neural network research, characteristics of neural networks terminology	7
2	**NEURAL NETWORKS MODELS** Models of neuron McCulloch – Pitts model, Perceptron, Adaline model, Basic learning laws, Topology of neural network architecture, Multilayer Neural Networks	5
3	**LEARNING METHODS AND PROPAGATION** Learning Methods, Backpropagation, Counterpropagation, ART, BAM, Associative memories	6
4	**INTRODUCTION TO FUZZY LOGIC** Fuzzy sets, Fuzzy model, Fuzzy rule generation Fuzzy inference system, Defuzzification	6
5	**GENETIC ALGORITHMS** Overview, Problem solving using GA, Implementation of GA and GP	5
6	**NEURO-FUZZY SYSTEMS** Introduction, Architecture of a Neuro-Fuzzy system and its applications	4
7	**MACHINE LEARNING:** Supervised learning: Primitive algorithms, Generative algorithms, Support Vector Machine, Ensemble methods. Unsupervised learning: K-means, Principal component analysis, Independent component analysis. Reinforcement learning and control.	6
8	**APPLICATIONS** Applications of GA & GP, Hybrid systems	3

List of Experiments:		**Number of Turns**
1	Implement OR, AND Using Perceptron in MATLAB Command-line Argument"	2
2	Implement OR, AND Using Perceptron in MATLAB GUI	2
3	Implement OR, AND, X-OR gate, Using back propagation algorithm in MATLAB using Command-line Argument as well as GUI.	2
4	Solve a given problem-1 (Operations) using Fuzzy Logic in MATLAB.	2
5	Solve a given problem-1 (Max-Min Composition) using Fuzzy Logic in MATLAB.	2
6	To find the solution of the function Maximize, given the constraints using GA approach.	1
7	Solve a given problem-1using using Fuzzy Logic in MATLB GUI	1
8	Study GA tool in MATLAB.	1

Course Outcomes: At the end of the course, students will be able to:	
1	Identify and describe soft computing techniques and their roles in building intelligent machines
2	Recognize the feasibility of applying a soft computing methodology for a particular problem
3	Effectively use existing software tools to solve real problems using a soft computing approach
4	Evaluate and compare solutions by various soft computing approaches for a given problem.

Suggested Books:

Sr. No.	Name of Book/ Authors/ Publisher	Year of Publication/ Reprint
1	Neuro fuzzy and soft computing by Jang, Pearson Education	1996
2	Learning and Soft Computing by Kecman, Pearson Education	2001
3	Fuzzy Sets and Fuzzy Logic - Klir and Yuan, PHI	1995
4	Neural Network in computer Intelligence by Fu, TMH	2003
5	Bio-Inspired Artificial Intelligence – Dario Floreano, PHI	2008

Course Name	**:**	**DIGITAL IMAGE PROCESSING**
Course Code	**:**	**CSN404**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:

The students should be able to learn the basic theory and algorithms that are widely used in digital image processing. They should learn current technologies and issues that are specific to image processing systems. They should develop hands-on experience to process images.

Total No. of Lectures – 42

Lecture wise breakup		Number of Lectures
1	**INTRODUCTION AND FUNDAMENTAL TO DIGITAL IMAGE PROCESSING** Origin of Digital Image Processing, Fundamental steps in Digital Image Processing, Components of Digital Image Processing System, Image sensing and acquisition, Image sampling, quantization and representation, Basic relationship between pixels.	7
2	**IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN & FREQUENCY DOMAIN** Basic grey level transformation, Histogram processing, Basics of Spatial filtering, Smoothing and Sharpening spatial filters, Introduction to Fourier Transform and the Frequency Domain, Discrete Fourier Transform, Smoothing and Sharpening Frequency – Domain filters.	8
3	**IMAGE RESTORATION** Image Degradation/Restoration Process, Noise models, restoration in presence of noise, Inverse filtering, Minimum Mean Square Filtering, Geometric menu filter, Geometric transformations.	5
4	**COLOR IMAGE PROCESSING** Color Fundamentals, Color models, Basis of full color image processing, Color transformations.	3
5	**IMAGE COMPRESSION** Fundamentals, Image compression models, Error free compression, Lossy compression.	5
6	**IMAGE SEGMENTATION** Detection of Discontinuities, Edge linking and boundary detection, Threshold, Region oriented segmentation.	5
7	**REPRESENTATION, DESCRIPTION AND RECOGNITION** Representation-chain codes, polygonal approximation and skeletons, Boundary descriptors-simple descriptors, shape numbers, Regional descriptors, topological descriptors.	3
8	**PATTERN RECOGNITION** Pattern and pattern classes, Tree classifiers: Decision trees, random forests, Parametric techniques: Maximum likelihood Estimation, Non-Parametric techniques: Kernel Density estimators, Nearest Neighbour methods.	6

List of Experiments:	Number of

		Turns
1	Implement the basic commands/ functions of an image processing tool.	1
2	Take an input image and plot its histogram with various ways as imhist, bar, stem, plot and Prove that histogram processing can be used for image enhancement.	1
3	Filtering using MATLAB package	1
4	Filtering for Blurring and Sharpening the image	2
5	Implement various Nonlinear Spatial Filters.	2
6	Implement various types of filters to remove the noise in an image.	2
7	Implement image compression algorithms.	2
8	Design problems related to image segmentation	1
9	Design problems related to image recognition, pattern recognition	1

Course Outcomes: At the end of the course, students will be able to:	
1	Develop simple image processing applications.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Digital Image processing By Rafael C. Gonzalez and Richard E. Woods- Pearson Education	2006
2	Digital Image Processing by A.K. Jain, PHI	1995
3	Digital Image processing (An algorithmic approach) By Madhuri A. Joshi - PHI	2006

Course Name	**:**	**CLOUD COMPUTING**
Course Code	**:**	**CSN405**
Credits	**:**	**4**
L T P	**:**	**3 1 0**

Course Objectives:

Students should be able to understand the concepts, processes, and best practices needed to successfully secure information within Cloud infrastructures. They should learn the basic Cloud types and delivery models. They should develop an understanding of the risk and compliance responsibilities and Challenges for each Cloud type and service delivery model.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION TO CLOUD COMPUTING** Online Social Networks and Applications, Cloud introduction and overview, Different clouds, Risks, Novel applications of cloud computing	**3**
2	**CLOUD COMPUTING ARCHITECTURE** Introduction Cloud computing architecture, On Demand Computing Virtualization at the infrastructure level, CPU Virtualization, Discussion on Hypervisors Storage Virtualization, The SPI Framework for Cloud Computing, The Cloud Services Delivery Model	**7**
3	**CLOUD DEPLOYMENT MODELS** Key Drivers to Adopting the Cloud, The Impact of Cloud Computing on Users, Governance in the Cloud, Barriers to Cloud Computing Adoption in the Enterprise	**3**
4	**SECURITY ISSUES IN CLOUD COMPUTING** Security in cloud computing environment, Infrastructure Security: The Network Level, The Host Level, The Application Level, Data Security and Storage, Aspects of Data Security, Data Security Mitigation Provider Data and Its Security,	**7**
5	**IDENTITY AND ACCESS MANAGEMENT**	**5**

	Trust Boundaries and IAM, IAM Challenges, Relevant IAM Standards and Protocols for Cloud Services, IAM Practices in the Cloud, Cloud Authorization Management	
6	**SECURITY MANAGEMENT IN THE CLOUD** Security Management Standards, Security Management in the Cloud, Availability Management: SaaS, PaaS, IaaS	**5**
7	**PRIVACY ISSUES** Privacy Issues, Data Life Cycle, Key Privacy Concerns in the Cloud, Protecting Privacy, Changes to Privacy Risk Management and Compliance in Relation to Cloud Computing, Legal and Regulatory Implications, U.S. Laws and Regulations, International Laws and Regulations	**5**
8	**AUDIT AND COMPLIANCE** Internal Policy Compliance, Governance, Risk, and Compliance (GRC), Regulatory/External Compliance, Cloud Security Alliance, Auditing the Cloud for Compliance, Security-As-a-[Cloud]	**7**

List of Experiments:		**Number of Turns**
1	Installation & Configuration of Oracle Virtual box for Windows XP & Android.	2
2	Installation Configuration of Hadoop.	2
3	Using Hadoop for counting word frequency with map reduce.	1
4	Service deployment research & uses over cloud- Google app & Amazon web services.	2
5	Mobile App Development using Google app & Amazon web services	4
6	Performance Evaluation of Services over cloud- Google App &Amazon Web Services.	1
7	Design and deploy a Private Cloud using Open Source Tools	1

Course Outcomes: At the end of the course, students will be able to:	
1	Identify security aspects of each cloud model
2	Develop a risk-management strategy for moving to the Cloud
3	Implement a public cloud instance using a public cloud service provider
4	Apply trust-based security model to different layers in the infrastructure stack
5	Distinguish between cloud providers and 3rd party managed service providers

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Cloud Computing Explained: Implementation Handbook for Enterprises, John Rhoton, Publication Date	2009
2	Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice), Tim Mather, ISBN-10: 0596802765,O'Reilly Media	2009
3	Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, Publisher: O'Reilly Media; ISBN-10: 0596156367, ISBN-13: 978-0596156367	2009
4	Cloud Computing Bible by Barrie Sosinsky Wiley Publication, ISBN-10: 0470903562	2011
5	Introduction to Cloud Computing by Timothy Chou	2010

Course Name	**:**	**AGILE SOFTWARE DEVELOPMENT**
Course Code	**:**	**CSN406**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:

The students should be able to understand the background and driving forces for taking an Agile approach to software development. They should understand the business value of adopting Agile approaches. They should understand the Agile development practices.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**FUNDAMENTALS OF AGILE** The Genesis of Agile, Introduction and background, Agile Manifesto and Principles, Overview of Scrum, Extreme Programming, Feature Driven development, Lean Software Development, Agile project management, Design and development practices in Agile projects, Test Driven Development, Continuous Integration, Refactoring, Pair Programming, Simple Design, User Stories, Agile Testing, Agile Tools**.**	**8**
2	**AGILE SCRUM FRAMEWORK** Introduction to Scrum, Project phases, Agile Estimation, Planning game, Product backlog, Sprint backlog, Iteration planning, User story definition, Characteristics and `content of user stories, Acceptance tests and Verifying stories, Project velocity, Burn down chart, Sprint planning and retrospective, Daily scrum, Scrum roles – Product Owner, Scrum Master, Scrum Team, Scrum case study, Tools for Agile project management.	**10**
3	**AGILE TESTING** The Agile lifecycle and its impact on testing, Test-Driven Development (TDD), xUnit framework and tools for TDD, Testing user stories - acceptance tests and scenarios, Planning and managing testing cycle, Exploratory testing, Risk based testing, Regression tests, Test Automation, Tools to support the Agile tester	**10**
4	**AGILE SOFTWARE DESIGN AND DEVELOPMENT** Agile design practices, Role of design Principles including Single Responsibility Principle, Open Closed Principle, Liskov Substitution Principle, Interface Segregation Principles, Dependency Inversion Principle in Agile Design, Need and significance of Refactoring, Refactoring Techniques, Continuous Integration, Automated build tools, Version control	**7**
5	**INDUSTRY TRENDS** Market scenario and adoption of Agile, Agile ALM, Roles in an Agile project, Agile applicability, Agile in Distributed teams, Business benefits, Challenges in Agile, Risks and Mitigation, Agile projects on Cloud, Balancing Agility with Discipline, Agile rapid development technologies	**7**

List of Experiments:		**Number of Turns**
1	Understand a given business scenario and identify product backlog, user stories and sprint tasks	2
2	Define user stories for a given feature	2
3	Fill user stories, sprint schedule and sprint tasks in an Agile tool such as AgileFant	2
4	Write unit tests aligned to xUnit framework for TDD	2
5	Refactor a given design for next sprint requirements	2
6	Execute continuous integration using a tool such as Jenkins	2
7	Automate a set of given tests using Test automation tool	1

Course Outcomes: At the end of the course, students will be able to:	
1	Understand the background and core practices for taking an Agile approach to software development
2	Understand the significance of value-driven delivery and continuous customer and user feedback in increasing team effectiveness
3	Have a guidance and decision making framework for self-organising Agile teams

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/**

		Reprint
1	Succeeding with Agile: Software development using Scrum, Mike Cohn, Pearson Education.	2010
2	Agile Software Development with Scrum, 1st Edition, Ken Schwaber	2014
3	Agile Software Development with Scrum By Ken Schawber, Mike Beedle, Pearson.	2008
4	Agile Software Development, Principles, Patterns and Practices By Robert C. Martin , Prentice Hall**.**	2002
5	Agile Testing: A Practical Guide for Testers and Agile Teams, By Lisa Crispin, Janet Gregory, Addison Wesley.	2008
6	Agile Software Development: The Cooperative Game By Alistair Cockburn, Addison Wesley	2006
7	User Stories Applied: For Agile Software By Mike Cohn, Addison Wesley.	2004

Course Name	**:**	**Natural Language Processing**
Course Code	**:**	**CSN407**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
The students should be able to study language and the tools that are available to efficiently study and analyze large collections of text. They should learn about and discuss the effects of electronic communication on our language.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION** A computational framework for natural language, description of English or an Indian language in the frame work, lexicon, algorithms and data structures for implementation of the framework, Finite state automata, The different analysis levels used for NLP (morphological, syntactic, semantic, pragmatic, Recursive and augmented transition networks. Applications like machine translations.	**7**
2	**WORD LEVEL AND SYNTACTIC ANALYSIS** Word Level Analysis: Regular Expressions, Finite-State Automata, Morphological Parsing, Spelling Error Detection and correction, Words and Word classes, Part-of Speech Tagging. Syntactic Analysis: Context-free Grammar, Constituency, Parsing-Probabilistic Parsing. Machine-readable dictionaries and lexical databases , RTN, ATN.	**8**
3	**SEMANTIC ANALYSIS** Semantic Analysis: Meaning Representation, Lexical Semantics, Ambiguity, Word Sense Disambiguation. Discourse Processing: cohesion, Reference Resolution, Discourse Coherence and Structure. Knowledge Representation, reasoning.	**10**
4	**NATURAL LANGUAGE GENERATION** Natural Language Generation (NLG): Architecture of NLG Systems, Generation Tasks and Representations, Application of NLG. Machine Translation: Problems in Machine Translation, Characteristics of Indian Languages, Machine Translation Approaches, Translation involving Indian Languages.	**10**
5	**INFORMATION RETRIEVAL AND LEXICAL RESOURCES** Information Retrieval: Design features of Information Retrieval Systems, Classical, Non-classical, Alternative Models of Information Retrieval, valuation Lexical Resources: World Net,Frame Net, Stemmers, POS Tagger.	**7**

List of Experiments:		**Number of Turns**
1	Implement program to perform automatic word analysis	1
2	Implement program to perform word generation	3

3	Implement programs related to morphology, N-Grams, N-Grams Smoothing	3
4	Implementation of Hidden Markov Models	3
5	Program to build POS Tagger, Chunker	3

Course Outcomes: At the end of the course, students will be able to:	
1	Learn natural language processing with manual and automated approaches.
2	Learn computational frameworks for natural language processing.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Natural Language understanding by James Allen, Pearson Education	2008
2	NLP: A Paninian Perspective by Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal, Prentice Hall	1995
3	Meaning and Grammar by G. Chirchia and S. McConnell Ginet, MIT Press	2000
4	An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition by Daniel Jurafsky and James H. Martin, Pearson Education	2008
5	Natural language processing in Prolog by Gazdar, & Mellish, Addison-Wesley	1989

Course Name	**:**	**SOFTWARE PROJECT MANAGEMENT**
Course Code	**:**	**CSN408**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
Students should be able to discuss various aspects of project management. To understand the tasks (organizing, planning and controlling) in software project management. To gain the ability to select tools and methodologies for various projects.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION TO SOFTWARE PROJECT MANAGEMENT** The characteristics of software projects, Reasons for IT project failure, Objectives of project management: time, cost and quality, Basics of Project Management, Stakeholders, Stages of Project, Activities Covered By Software Project Management, Project and Product Life Cycles, Project Management Knowledge areas, Project success factors, role of project manager.	**5**
2	**PROJECT EVALUATION AND PLANNING** Strategic Assessment, Cost Benefit Analysis and techniques, earned value analysis, Understanding organizations, stakeholder's management, project phases & project life cycles, Objectives of project planning, Project schedule, Iterative steps for planning, Project Plan documentation methods, Software Requirement Specification, Types Of Risk, Managing Risk, Risk Planning And Control.	**6**
3	**MONITORING AND CONTROL** Visualizing Progress, Cost Monitoring, Plan versus Control, managing the plan, Change Control, The Deadline Effect, Reviews, feedback and reporting mechanisms, revisiting the plan.	**3**
4	**PROJECT SCOPE MANAGEMENT** Scope Planning & Scope management plans, Function point calculation, Scope definitions & project scope statement, Work Breakdown Structure (WBS), WBS dictionary, scope verification, scope change control.	**3**

5	**PROJECT TIME AND COST MANAGEMENT** Development of project schedule, activities sequencing and their dependencies, network diagrams, activity recourse estimation, activity duration estimation, schedule development, Gantt Charts, Critical path method, Programme evaluation & review technique (PERT) and CPM, concept of slack time, schedule control, Basic principles of cost management, Cost estimating, type of cost estimate, COCOMO, cost budgeting, cost control, earned value management.	**6**
6	**PROJECT RISK MANAGEMENT** Risk Management planning, sources of risk, risk identification, Reactive V/S proactive Risk Strategies, risk register, qualitative risk analysis, decision trees & expected monetary value, simulation, risk response planning, risk refinement, risk mitigation, risk monitoring & control, Risk assessment.	**3**
7	**PROJECT QUALITY MANAGEMENT** Quality Planning, quality Assurance, Quality control, Tool & techniques for quality control, Formal approaches to SQA, Pareto Analysis, Six Sigma, CMMI, ISO Standards, configuration management, Defect Prevention Planning.	**4**
8	**PROJECT HUMAN RESOURCE MANAGEMENT** Human resource planning, project organizational charts, responsibility assignment metrics, acquiring project team, resource assignment, resource loading, resource leveling, Different team structures developing project teams.	**3**
9	**PROJECT COMMUNICATION MANAGEMENT** Communication Planning, project metrics, Performance reporting, managing stakeholders, Improving project communication, Performance reporting.	**3**
10	**PROJECT PROCUREMENT MANAGEMENT** Procurement management plans, contracting- contract statement of work, planning contracts, requesting seller responses, selecting sellers, administrating the contract, closing the contract, outsourcing of products.	**3**
11	**SOFTWARE CONFIGURATION MANAGEMENT:** Why versions exist, why retain versions, SCI, Releases vs. version, Change Control and Management.	**3**

List of Experiments:		**Number of Turns**
1	Using Function Point calculation tools for estimation	1
2	Implementation of various exercises using PERT, CPM methods	2
3	Preparing Project Plan for a Software Project for Lab Project or case study.	2
4	SCM/ Change management	2
5	Exposure to Project management tool	3
6	Minor Project with project management activities	3

Course Outcomes: At the end of the course, students will be able to:	
1	Understand and practice the process of project management and its application in delivering successful Software projects;
2	Demonstrate the ability to structure a problem, evaluate and analyze the project work.
3	Monitor the progress of a project, revising targets or countract drift.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Bob Hughes, Mike Cotterell and Rajib Mall, Software Project Management, Third Edition, Tata McGraw-Hill.	2009
2	Gopalaswamy Ramesh, Managing global software Projects: How to lead geographically distributed Teams, Manage Processes and Use Quality Models, Tata McGraw-Hill.	2006
3	Project Management Body Of Knowledge(PMBOK).	2000

4	Pankaj Jalote, Software Project Management in Practice, Pearson Education.	2004
5	S.A. Kelkar, Software Project Management, A Concise Study, Revised Edition, Prentice-Hall India	2003
6	Elaine Marmel, Microsoft Office Project 2003 Bible, Wiley Publishing Inc.	2004
7	Kathy Schwalbe; Information Technology Project Management fourth edition, Thomson Course Technology.	2006

Course Name	**:**	**BIG DATA ANALYTICS**
Course Code	**:**	**CSN409**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
The students should be able to understand and apply big data flow to actual projects as well as apply data analytics life cycle to big data projects. The student should identify and successfully apply appropriate techniques and tools to solve big data problems.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION TO BIG DATA** Introduction to BigData Platform, Traits of Big data, Challenges of Conventional Systems, Web Data, Evolution of Analytic Scalability, Analysis vs Reporting, Statistical Concepts: Sampling Distributions, Re-Sampling, Statistical Inference, Prediction Error.	**6**
2	**BASIC DATA ANALYSIS AND DATA ANALYTIC METHODS USING R** Regression Modelling, Multivariate Analysis, Bayesian Modelling, Inference and Bayesian Networks, Support Vector and Kernel Methods, Analysis of Time Series: Linear Systems Analysis, Nonlinear Dynamics, Rule Induction, Neural Networks: Learning And Generalization, Competitive Learning, Principal Component Analysis and Neural Networks, Fuzzy Logic: Extracting Fuzzy Models from Data Fuzzy Decision Trees, Stochastic Search Methods. Introduction to R, Statistics for Model Building and Evaluation.	**10**
3	**FREQUENT ITEMSETS AND CLUSTERING** Mining Frequent Itemsets: Market Based Model, Apriori Algorithm, Handling Large Data Sets in Main Memory, Limited Pass Algorithm, Counting Frequent Itemsets in a Stream, Clustering Techniques: Hierarchical, K-Means, Frequent Pattern based Clustering Methods.	**8**
4	**MINING DATA STREAMS** Introduction to Streams Concepts: Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream: Filtering Streams, Counting Distinct Elements in a Stream, Estimating Moments, Counting Oneness in a Window, Decaying Window, Real time Analytics Platform(RTAP) Applications, Case Studies, Real Time Sentiment Analysis, Stock Market Predictions.	**10**
5	**FRAMEWORK, TECHNOLOGIES, TOOLS AND VISUALIZATION** MapReduce: Hadoop, Hive, MapR, Sharding, NoSQL Databases: S3, Hadoop Distributed File Systems, Visualizations: Visual Data Analysis Techniques, Interaction Techniques; Systems and Analytics Applications, Analytics using Statistical packages, Industry challenges and application of Analytics.	**8**

List of Experiments:		**Number of Turns**
1	Hands-on with Map Reduce: Hadoop, Hive, MapR	4
2	Hands-on with NoSQL Databases: S3, Hadoop Distributed File System(HDFS)	3
3	Hands-on with Statistical Packages	3
4	Hands-on with Visual Data Analysis tools	3

Course Outcomes: At the end of the course, students will be able to:	
1	Explain the statistics of Big Data Analytics.
2	Describe and use various analytical methods.
3	Analyze data in real applications and design efficient mining techniques
4	Analyze various applications on tools like MapReduce, Hadoop, and S3.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Bart Baesens, "Analytics in a Big Data World: The Essential Guide to data Science and its Applications", Wiley publications.	2014
2	Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer.	2007
3	Anand Rajaraman and Jeffrey David Ullman, "Mining of Massive Datasets", Cambridge University Press.	2012
4	Bill Franks, "Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics", John Wiley & sons	2012
5	Glenn J. Myatt, "Making Sense of Data", John Wiley & Sons.	2007

Course Name	**:**	**BIOINFORMATICS**
Course Code	**:**	**CSN410**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
The students should be able to understand the scope of Bioinformatics. They should know about popular bioinformatics databases and sequence alignment algorithms.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION** History, scope and important contributions, aims and tasks of Bioinformatics, applications of Bioinformatics, challenges and opportunities, introduction to NCBI data model, various file formats for biological sequences.	**6**
2	**BIOLOGICAL DATABASES** Importance of databases, biological databases, primary sequence databases, composite sequence databases, secondary databases, nucleic and sequence databases, protein sequence databases, structure databases, bibliographic databases, specialized genomic resources, analysis packages	**7**
3	**DATABASE SEARCH METHODS** Methods for searching sequence databases like FASTA and BLAST algorithms, Statistical analysis and evaluation of BLAST results.	**7**
4	**SEQUENCE COMPARISON METHODS** Methods for comparison of two sequences, Needleman Wush and Smith Waterman algorithms. Analysis of computational complexities, merits and demerits of these algorithms, theory of scoring matrices and their use for sequence comparison.	**8**
5	**SEQUENCE ALIGNMENT METHODS** Sequence analysis of biological data, significance of sequence alignment, pair wise sequence alignment methods, use of scoring matrices and gap penalties in sequence alignments, multiple sequence alignment methods, tools and applications of multiple sequence	**8**

	alignment.	
6	**PREDICTIVE METHODS USING DNA AND PROTEIN SEQUENCES** Gene prediction strategies, protein prediction strategies, molecular visualization tools, phylogenetic analysis: concept of trees, phylogenetic trees and multiple alignments.	**6**

List of Experiments:		**Number of Turns**
1	Hands-on with Nucleic acid databases (NCBI, DDBJ, EMBL), Protein databases (Primary, Composite and Secondary)	2
2	Hands-on with Specialized Genome databases (SGD, TIGR, ACeDB), Structure databases (CATH, SCOP, PDBsum)	2
3	Hands-on with methods for searching sequence databases	3
4	Hands-on with sequence comparison and sequence alignment methods	3
5	Hands-on with predictive methods	3

Course Outcomes: At the end of the course, students will be able to:	
1	Search various biological sequence databases
2	Perform sequence comparison
3	Perform sequence alignment
4	Apply predictive methods on DNA and protein sequences

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Andreas D Baxevanis & B F Francis, "Bioinformatics-A practical guide to analysis of Genes and Proteins", John Wiley	2010
2	T K Attwood, D J Parry-Smith, " Introduction to Bioinformatics", Pearson Education	2005
3	Neil C. Jones, Pavel A. Pevzner, "An introduction to Bioinformatics Algorithms", MIT Press	2005
4	Gary Benson Roderic, "Algorithms in Bioinformatics", Springer	2004

Course Name	**:**	**NETWORK SECURITY**
Course Code	**:**	**CSN411**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
The students should be able to investigate core security technologies and security policies to mitigate risks. They should gain an understanding of network perimeter security design principles. They should also gain an understanding of free/ commercial security tools and their applications and develop the security solution for a given application/scenario.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**PACKET FILTERING** Packet Filtering Principle, TCP and UDP Ports, TCP's Three-way Handshake, Router as a Packet Filter, An Alternative Packet Filter: IP Chains, Egress Filtering, Industry standard protocols: working of router protocol.	**5**
2	**STATEFUL & PROXY FIREWALLS** Working of Stateful Firewall, The Concept of State, Stateful Filtering and Stateful Inspection, Fundamentals of Proxying, Pros and Cons of Proxy Firewalls, Types of Proxies,	**5**

	Tools for Proxying	
3	**SECURITY POLICY** Firewalls Policy, Design of a security policy, Perimeter Considerations	**2**
4	**VIRTUAL PRIVATE NETWORKS** VPN Basics, Advantages and Disadvantages of VPNs, IPSec Basics	**3**
5	**NETWORK INTRUSION DETECTION & PREVENTION SYSTEMS** Network Intrusion Detection Basics, the Roles of Network IDS in a Perimeter Defense, IDS Sensor Placement, IPS, IPS Limitations, NIPS, Host-Based Intrusion Prevention Systems, Case Studies	**4**
6	**HOST HARDENING & DEFENSE COMPONENTS** Need for Host Hardening, Removing, Disabling or Limiting Access of Unnecessary Programs, Data and Configuration Files, Controlling User and Privileges, hardening hosts and the Perimeter, Antivirus Software, Host-Based Firewalls, Host-Based Intrusion Detection, Challenges of Host Defense Components, Preventing TCP/UDP exploits from DoS attacks.	**8**
7	**DESIGNING A SECURE NETWORK PERIMETER** The Role of a Router, The Router as a Perimeter & Security Device, Router Hardening, Fundamentals of Secure Perimeter Design, Gathering Design Requirements, Design Elements for Perimeter Security, Separating Resources, Security Zones, Common Design Elements, VLAN-Based Separation	**7**
8	**MAINTAINING A SECURITY PERIMETER** System and Network Monitoring, Incident Response, Accommodating Change	**4**
9	**NETWORK LOG ANALYSIS** The Importance of Network Log Files, Log Analysis Basics, Analyzing Router Logs, Analyzing Network Firewall Logs, Analyzing Host-Based Firewall and IDS Logs	**4**

List of Experiments:		**Number of Turns**
1	Use of Cain and Abel Tool Cracking the system password by using brute force method. Cracking the system password by using dictionary attack. Sniff the password of user's account on the network. Getting the list of all wireless access points which are security enabled and which are not.	**8**
2	Email Tracing	**2**
3	Consider any 4 apps of mobile phone and compare their security permissions and personal information being accessed.	**3**

Course Outcomes: At the end of the course, students will be able to:	
1	Explain fundamental concepts of network vulnerabilities and attacks.
2	Demonstrate the skill to penetrate service vulnerability.
3	Implement, monitor and maintain a secure network consisting of enterprise level routers and switches
4	Understand the role of AAA and IPSec in securing networks.
5	Understand how to design and implement firewall technologies that complement network policies in securing the perimeter of a network
6	Learn to design/develop/ implement the security solution for a given application.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Inside Network Perimeter Security, Second Edition, Stephen Northcutt; Lenny Zeltser; Scott Winters; Karen Kent; Ronald W. Ritchey, Sams	2005
2	Network Perimeter Security: Building Defense In-Depth , Cliff Riggs, Proteris Group, Waterbury, Vermont, USA	2003
3	W. Stallings, Network Security Essentials (3rd Edition), Prentice-Hall,	2007

4	W. R. Stevens, TCP/IP Illustrated, Vol. 1: The Protocols, Addison-Wesley	1993
5	D. E. Comer, Internetworking with TCP/IP, Vol.1 (4th Edition), Prentice Hall,	2000
6	R. Oppliger, Internet and Intranet Security (2nd edition), Artech House	2002
7	W. R. Cheswick and S.M. Bellovin, Firewalls and Internet security (2nd edition), Addison-Wesley,	2003

Course Name	**:**	**APPLIED CRYPTOGRAPHY**
Course Code	**:**	**CSN412**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
The students should be able to understand basic number theory, with a focus on computational aspects and applications in cryptography. They should understand basic design principals of symmetric and asymmetric cryptography. They should learn how cryptanalytic attacks work and thereby how to avoid common design flaws.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**FOUNDATIONS** Substitution Ciphers and Transposition Cipher, Block cipher, Stream cipher.	**4**
2	**CRYPTOGRAPHIC PROTOCOLS** Introduction to Protocols, Communications using Symmetric Cryptography, One-Way Functions, Communications using Public-Key Cryptography, Digital Signatures, Digital Signatures with Encryption, Random and Pseudo-Random Sequence Generation, Basic Protocols: Key Exchange, Authentication, Authentication And Key Exchange, Formal Analysis Of Authentication And Key-Exchange Protocols, Multiple-Key Public-Key Cryptography.	**8**
3	**CRYPTOGRAPHIC TECHNIQUES** **Key Length & Management:** Symmetric Key Length, Public-Key Key Length, Comparing Symmetric And Public-Key Key Length, **,**Generating Keys, Nonlinear Keyspaces, Transferring Keys, Verifying Keys, UPDATING KEYS, Storing Keys, Backup Keys.	**10**
4	**CRYPTOGRAPHIC ALGORITHMS** Mathematical Theory, NUMBER THEORY, FACTORING, Prime Number Generation, Discrete Logarithms in a Finite Field, Data Encryption Standard: Description of DES, Security of DES, Differential And Linear Cryptanalysis, Design Criteria, DES Variants, DES modes of operation, Other Stream Ciphers and One-Way Hash Functions RC4, One-Way Hash Functions, MD5, Secure Hash Algorithm (SHA), Message Authentication Codes	**8**
5	**PUBLIC-KEY ALGORITHMS** Background, RSA, Elliptic Curve Cryptosystems, Digital Signature Algorithm, Key-Exchange Algorithms: DIFFIE-HELLMAN	**8**
6	**IMPLEMENTATIONS** PRETTY GOOD PRIVACY (PGP), SMART CARDS	**4**

List of Experiments:		**Number of Turns**
1	Install JCRYPT tool (or any other equivalent) and demonstrate Asymmetric, symmetric crypto algorithm, hash and digital/PKI Signatures.	**3**
2	Using Wireshark / Ethereal Protocol Analyser • How to capture the network's traffic. • How to save the filtered traffic. • Saving the traffic in file in pcap or in log format.	**2**

	• Study and analyzing the captured packets. • Creating a case where the server and client had a conversation. • Eavesdropping the particular IP in the network.	
3	Frequency Analysis This lab will introduce students to frequency analysis, a method used to decode ciphertext by studying the frequency of letters.	**2**
4	Hash Function This lab will introduce students to hash functions and how they provide for message integrity. Students will be asked to use hashing to detect if an encrypted message has been tampered with. Students will also need to show that this integrity check can be bypassed by tampering with both the ciphertext and the hash code.	**2**
5	Hands On: Some sample Problems which can be given to the students • How to Ensure the validity of Forensic Evidence by Using a Hash Function? • How to extract evidence from a Mobile Phone? • How to use Open SSL to generate digital Signatures • Conduct Performance Analysis between various Symmetric Algorithms	**4**

Course Outcomes: At the end of the course, students will be able to:	
1	Apply the basic rules of public key and symmetric encryption for practical cryptographic problems.
2	Demonstrate the design and use of hash functions, digital signatures, and key distribution with a wide range of key types.
3	Design and implement a cryptography algorithm.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Applied Cryptography protocols, algorithms, and source code in C, Second Edition, Bruce Schneier, John Wiley & Sons	1996
2	Handbook of Applied Cryptography, by Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, CRC Press ISBN: 0-8493-8523-7.	1996

Course Name	**:**	**COMPUTER CRIME INVESTIGATION AND FORENSICS**
Course Code	**:**	**CSN413**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
The students should be able to have technical skills and competencies in the field of forensic computing. They should be able to protect the computer system during the forensic examination from any possible alteration, damage, data corruption, or virus introduction. They should understand how information is stored and used on digital devices.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION** Types of computer crime, history, surveys, statistics and global connections, Aspects of Cyber Warfare and Cyber Terrorism, Dynamic, Human and Technical Aspects of Cyber Warfare and Cyber Terrorism, Identification, Authorization and Access Control.	**7**
2	**SOCIAL ENGINEERING** Mail Bombs, Bug Exploits, Stalking, Spam, Phishing and Pharming	**4**

3	**MALWARE** The types, effects and investigations, DoS and Distributed DoS: the causes, mechanisms, case studies and counter-measures.	**7**
4	**NETWORK CRIMES** Unauthorized access to computers, computer intrusions, white collar crimes, viruses and malicious code, Internet hacking and cracking, Hacking methodologies via Internet and attacks to other networks, Virus attacks.	**4**
5	**COMPUTER FORENSICS & INVESTIGATION** Preparation of Investigation, Procedures, Understanding Data Recovery, Data Acquisition, Processing Crime & Incident scenes, Current Computer Forensic tools, Computer Forensic Analysis & Validations, recovery Graphic Files, Network Forensics, Email Investigations, Mobile Device Forensics	**6**
6	**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, Audio Video Analysis, Windows System Forensics, Linux System Forensics, Network Forensics.	**4**
7	**LEGAL MEASURES** Computer Misuse, Criminal Damage, Software Piracy, Forgery, Investigative Powers	**4**
8	**CASE STUDIES** Investigations into hacking, cases and PC misuse, Investigations, Incident Handling and Forensic Examination, The Future: The expansion of the Internet, unsuitable material Identity Theft and Fraud	**6**

List of Experiments:		**Number of Turns**
1	Practical problems on Data and Evidence Recovery	2
2	Practical problems on Hacking Methodologies	2
3	Practical problems on Mobile Device Forensics	2
4	Practical problems on Windows Forensics	2
5	Practical using Cyber Forensic Investigation Tools	3
6	Practical problems on Web Browser Forensics and Email Tracing	2

Course Outcomes: At the end of the course, students will be able to:	
1	Demonstrate use of computer forensics tools and appropriate skills and knowledge to perform investigations
2	Analyze digital devices to establish user activity
3	Research the development of new devices and technologies and how current digital forensics methods will apply to them.
4	Gain insight knowledge to understand attack profiles, investigation tools and techniques
5	Gain ability to perform Critical analysis of data to identify evidence
6	Be able to trace malicious internet activity and analyze email trails.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Cyber Warfare and Cyber Terrorism , Andrew Colarik and Lech Janczewski, editors Dorothy E. Denning, ISBN13: 9781591409915.	2007
2	Computer Forensics and Investigations, Nelson, Phillips Enfinger, Steuart, CENGAGE Learning	2009
3	Information Warfare and Security, Addison-Wesley, Hedley & Aplin,	1999
4	Blackstone's Statutes on IT and E-Commerce, Oxford University Press, C. Stoll, The Cuckoo's Egg, Pan Book Publishers.	2002

Course Name	**:**	**BIOMETRIC SECURITY**
Course Code	**:**	**CSN414**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
The students should be able to understand a broad range of approaches to biometrics reflecting both fundamental principles and the current state-of-the-art practices. They should develop an understanding of the fundamental components common to all biometric systems.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**BIOMETRICS INTRODUCTION** Benefits of biometrics over traditional authentication systems, benefits of biometrics in identification systems, comparison of various biometric traits, selecting a biometric for system, Applications. Key biometric terms and processes, biometric verification and identification, how biometric matching works, Accuracy in biometric systems, Metrics for evaluating biometric systems: FAR, FRR, ERR etc.	**7**
2	**PHYSIOLOGICAL BIOMETRIC TECHNOLOGIES** Fingerprints: Technical description, characteristics, Competing technologies, strengths, weaknesses and deployment. Facial scan: Technical description, characteristics, weaknesses and deployment. Iris scan: Technical description, characteristics, strengths, weaknesses and deployment. Hand scan: Technical description, characteristics, strengths, weaknesses and deployment.	**15**
3	**BEHAVIORAL BIOMETRIC TECHNOLOGIES** Handprint Biometrics, Signature and handwriting technology: Technical description, classification, keyboard /keystroke dynamics, Voice: data acquisition, feature extraction, characteristics, strengths, weaknesses, deployment	**10**
4	**MULTI BIOMETRICS** Basic concept of Multi-modal biometric Systems, Advantages of Multimodal over Unimodal Biometric Systems, Multimodal fusion techniques.	**5**
5	**BIOMETRIC SECURITY MODALS** Introduction to Biometric Security Modals, Various attack vectors and there remedial solutions, Template Security techniques.	**5**

List of Experiments:		**Number of Turns**
1	Basis hands on practice on image processing toolbox in matlab	1
2	To Develop a fingerprint verification system with following stages a. Data Accusation. b. Image Enhancement c. Image Segmentation d. Minutia Extraction e. Fingerprint Alignment. f. Fingerprint Matching	2
3	Image Enhancement Techniques for Forensic Crime Scene Investigations or Latent prints	1
4	Implement atleast one of the Face recognition techniques like PCA, LDA and ICA	2
5	Design and implement the various stages for iris recognition like localization, segmentation, matching.	2
6	Design and implement the various stages for signature recognition such as segmentation, thinning etc.	2
7	Design and implement the various stages for voice recognition such as MFCC computations.	2
8	Testing and implementing one of the fusion strategies of multimodal biometrics such as score level, decision level etc.	1

Course Outcomes: At the end of the course, students will be able to:	
1	Understand Design and Implement Modern biometric technologies and the generic components of a biometric system.
2	Implement Pattern recognition and feature extraction in biometrics systems.
3	Select the most appropriate biometric for a given application.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Anil K. Jain, Michigan State University, USA, Patrick Flynn University of Notre Dame, USA, Arun A. Ross West Virginia University, USA , "Handbook of Biometrics".	2008
2	Biometrics, Identity Verification in a Networked World, By Samir Nanavati, Michael Thieme and Raj Nanavati , Wiley	2011
3	Anil K. Jain Michigan State University, E. Lansing, Michigan and Ruud Bolle and Sharath Pankanti IBM, T.J. Watson Research Center Yorktown Heights, New York Kluwer Academic ," Biometrics Personal Identification in Networked Society", Kluwer Academic Publishers New York, Boston, Dordrecht, London, Moscow.	2002
4	Biometrics for Network Security, By Paul Reid, Pearson	2012
5	Biometrics: Concepts and Applications By G.R. Sinha, Sandeep Patil, Wiley	2013

Course Name	**:**	**Advanced Computer Networks**
Course Code	**:**	**CSN415**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
The students should have a deeper insight into the advanced topics of computer networks and crucial protocols in computer networks. They should develop an understanding of the underlying structure of networks and how they operate as client/server architecture, network scalability. They should be able to analyze simple protocols and independently study literature concerning computer networks

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION** Overview of computer networks; seven-layer architecture; TCP/IP suite of protocols.	**3**
2	**MEDIUM ACCESS** MAC protocols for high-speed networks -Gigabit Ethernet,RPR etc.	**4**
3	**INTERNETWORKING AND ROUTING** Internetworking problem – packet switching; Delivery and Forwarding of IPpackets; Brief introduction to IPv4, scaling IP address space, Unicast routing protocols, IPv6 protocol	**7**
4	**RESOURCE MANAGEMENT** Data traffic – traffic descriptor, traffic profiles; Congestion – definition, network performance; Congestion control techniques – open loop congestion control and closed loop congestion control; Congestion Control in TCP.	**5**
5	**QUALITY OF SERVICE** Introduction; Flow characteristics; Flow classes; Techniques to improve QoS – Scheduling, Traffic shaping, Resource Reservation, Admission control; Internet QoS models – Integrated services, Differentiated services.	**4**
6	**GROUP COMMUNICATION** Introduction; IP Multicast addresses; IGMP – group management, messages, operation;	**6**

	Multicast Routing and Transport, Multicast routing protocols – MOSPF, DVMRP, CBT, PIM, MBONE.	
7	**TRANSPORT LAYER PROTOCOLS** TCP protocol dynamics; new options in TCP; brief introduction to SCTP, T/TCP.	**6**
8	**WIRELESS NETWORKS** Wireless PAN, LAN, MAN, WAN;Mobile network layer; Mobile transport layer.	**7**

List of Experiments:		**Number of Turns**
1	Implementation of various protocols	3
2	Programs related to resource management	3
3	Programs related to group communication	3
4	Projects	4

Course Outcomes: At the end of the course, students will be able to:	
1	Independently understand basic computer network technology.
2	Familiarize with MAC protocols for high speed networks.
3	Identify concepts underlying Internet protocol, their main characteristics and functionality and their extension.
4	Understand various traffic parameters and congestion control techniques.
5	Explain current QoS architectures and mechanisms, and the QoS support challenges in future networks.
6	Understand the principles and functionality of various multicast and transport protocols.
7	Understand the needs of wireless networks and some extensions to mobility at various layers.

Suggested Books:

Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Andrew Tenenbaum. Computer Networks, PHI	1993
2	Forouzan, Local Area Networks, TMH	2002
3	Forouzan, TCP/IP Protocols, TMH	2010
4	Forouzan, Data Communication and Networking, TMH	2013
5	Computer Networking, A Top-Down Approach Featuring the Internet - J. Kurose and K. Ross, 6th Ed. (Pearson).	2012
6	Schiller J., Mobile Communications, Addison Wesley.	2000
7	G. Keiser, "Local Area Networks", 2nd Edition, TMH.	2002
8	D. Bertesekas and R. Gallager, "Data Networks", 2nd Edition, PHI.	2000
9	William Stallings, "Data & Computer Communication", PHI, 6th Edition	2002

Course Name	**:**	**ADVANCED WIRELESS AND MOBILE NETWORKS**
Course Code	**:**	**CSN416**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:

The students should get familiar with the wireless/mobile market and the future needs and challenges. They should get familiar with key concepts of wireless networks, standards, technologies and their basic operations. They should learn how to design and analyze various medium access. They should learn how to evaluate MAC and network protocols using network simulation software tools.

Total No. of Lectures – 42

Lecture wise breakup		Number of Lectures
1	**INTRODUCTION** Wireless Networking Trends, Key Wireless Physical Layer Concepts, Multiple Access Technologies- CDMA, FDMA, TDMA, Spread Spectrum technologies, Frequency reuse, Radio Propagation and Modeling, Challenges in Mobile Computing: Resource poorness, Bandwidth, energy, etc.	**4**
2	**WIRELESS LOCAL AREA NETWORKS: IEEE 802.11** Wireless LANs Physical & MAC layer, 802.11 MAC Modes (DCF & PCF) IEEE 802.11 standards, Architecture & protocols, Infrastructure vs. Adhoc Modes, Hidden Node & Exposed Terminal Problem, Problems, Fading Effects in Indoor and outdoor WLANs, WLAN Deployment issues	**10**
3	**WIRELESS CELLULAR NETWORKS:** 1G and 2G, 2.5G, 3G, and 4G, Mobile IPv4, Mobile IPv6, TCP over Wireless Networks, Cellular architecture, Frequency reuse, Channel assignment strategies, Handoff strategies, Interference and system capacity, Improving coverage and capacity in cellular systems, Spread spectrum Technologies	**10**
4	WiMAX (Physical layer, Media access control, Mobility and Networking), IEEE 802.22 Wireless Regional Area Networks, IEEE 802.21 Media Independent Handover Overview	**3**
5	**WIRELESS SENSOR NETWORKS** Introduction, Application, Physical, MAC layer and Network Layer,Power Management, Tiny OSOverview	**4**
6	**WIRELESS PANs** Bluetooth AND Zigbee, Introduction to Wireless Sensors, Introduction to Vehicular Adhoc Networks	**3**
7	**SECURITY** Security in wireless Networks Vulnerabilities, Security techniques, Wi-Fi Security, IEEE 802.11x and IEEE 802.11i standards, DoS in wireless communication	**8**

List of Experiments:		Number of Turns
1	To get familiar with the following tools: Network Miner, SMAC, WEP key generator or Wepkey, IPscan, Netstumbler, xarp, wzcook, wireshark, Qualnet 4.5	2
2	To study and compare various channel models for wireless channels using Qualnet	2
3	To create a network topology, simulate it using NS2 and analyze the trace file.	1
4	To design different topologies using IEEE 802.11 and conduct Performance Analysis	2
5	To perform wireless path loss computations (Indoor & Outdoor)	1
6	Setup & Configuration of Wireless Access Point (AP), Setup and creation of an Adhoc Network, Setup and creation of Mobile Horspots and conduct Performance Analysis of the link	2
7	Practical study of WLAN : Ad Hoc & Infrastructure Mode, Study of Bluetooth Protocol and Applications, Wireless Network Security : kismet and Netstumbler	2
8	Simulations using NS3 for IEEE 802.11 (Wifi)	1

Course Outcomes: At the end of the course, students will be able to:	
1	Demonstrate advanced knowledge of networking and wireless networking and understand various types of wireless networks, standards, operations and use cases
2	Be able to design WLAN, WPAN, WWAN, Cellular based upon underlying propagation and performance analysis.
3	Demonstrate knowledge of protocols used in wireless networks and learn simulating wireless networks.
4	Design wireless networks exploring tradeoffs between wire-line and wireless links.
5	Develop mobile applications to solve some of the real world problems.

Suggested Books:		
Sr.	**Name of Book/ Authors/ Publisher**	**Year of**

No.		Publication/ Reprint
1	Schiller J., Mobile Communications, Addison Wesley	2000
2	Stallings W., Wireless Communications and Networks, Pearson Education	2005
3	Stojmenic Ivan, Handbook of Wireless Networks and Mobile Computing, John Wiley and Sons Inc	2002
4	Yi Bing Lin and Imrich Chlamtac, Wireless and Mobile Network Architectures, John Wiley and Sons Inc	2000
5	Pandya Raj, Mobile and Personal Communications Systems and Services, PHI	2004

Course Name	**:**	**WIRELESS SENSOR NETWORKS**
Course Code	**:**	**CSN417**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:

The students should be able to have an understanding of the fundamental concepts of wireless sensor networks and have a basic knowledge of the various protocols at various layers. They should learn the architecture of the sensor node and various operating systems used for sensor network design. They should learn to apply sensor network protocols, mechanisms, and algorithms to implement sensing systems.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION** Motivation for a Network of Wireless Sensor Nodes – Definitions and Background, Challenges and constraints; Applications of wireless sensor networks.	**3**
2	**NODE ARCHITECTURE** The Sensing Subsystem - Analog-to-Digital Converter; The Processor Subsystem, Communication Interfaces, Prototypes.	**3**
3	**OPERATING SYSTEMS** Functional Aspects, Nonfunctional Aspects, Prototypes	**4**
4	**PHYSICAL LAYER** Basic Components, Source Encoding, Channel Encoding, Modulation, Signal Propagation	**4**
5	**MEDIUM ACCESS CONTROL** Overview, Wireless MAC Protocols, Characteristics of MAC Protocols in Sensor Networks, Contention-Free MAC Protocols, Contention-Based MAC Protocols, Hybrid MAC Protocols	**4**
6	**NETWORK LAYER** Overview, Routing Metrics, Flooding and Gossiping, Data-Centric Routing, Proactive Routing, On-Demand Routing, Hierarchical Routing, Location-Based Routing, QoS-Based Routing Protocols	**4**
7	**TRANSPORT LAYER** Traditional Transport Control Protocols, Transport Protocol Design Issues, Examples of Existing Transport Control Protocols, Performance of Transport Control Protocols	**4**
8	**POWER MANAGEMENT** Local Power Management Aspects, Dynamic Power Management, Conceptual Architectu	**4**
9	**TIME SYNCHRONIZATION** Clocks and the Synchronization Problem, Time Synchronization in Wireless Sensor Networks, Basics of Time Synchronization, Time Synchronization Protocols	**4**
10	**LOCALIZATION** Overview, Ranging Techniques, Range-Based Localization, Range-Free Localization, Event-Driven Localization	**4**

11	**SENSOR NETWORK PROGRAMMING** Challenges in Sensor Network Programming, Node-Centric Programming, Macroprogramming, Dynamic Reprogramming, Sensor Network Simulators	**4**

List of Experiments:		**Number of Turns**
1	Implement simulation experiments and projects based on Physical layer and MAC layer of wireless sensor network	3
2	Implement simulation experiments and projects based on network layer of wireless sensor network	3
3	Implement simulation experiments and projects based on transport layer of wireless sensor network	3
4	Implement simulation experiments and projects based on application layer of wireless sensor network	4

Course Outcomes: At the end of the course, students will be able to:	
1	Describe and explain radio standards and communication protocols for wireless sensor networks.
2	Explain the function of the node architecture and use of sensors for various applications.
3	Explain operating systems and programming languages for wireless sensor nodes.
4	Be familiar with architectures, functions and performance of wireless sensor networks systems and platforms.
5	Describe and analyze the specific requirements for applications in wireless sensor networks regarding energy supply, memory, processing and transmission capacity.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	W. Dargie and C. Poellabauer, "Fundemntals of Wireless Sensor Networks – Theory and Practice", Wiley.	2010
2	KazemSohraby, Daniel Minoli and TaiebZnati, "wireless sensor networks - Technology, Protocols, and Applications", Wiley Interscience.	2007
3	Takahiro Hara,Vladimir I. Zadorozhny, and Erik Buchmann, "Wireless Sensor Network Technologies for the Information Explosion Era", springer.	2010
4	Edgar H. Callaway, "Wireless Sensor Networks: Architectures and Protocols", CRC press.	2003
5	C. S. Raghavendra, Krishna M. Sivalingam and TaiebZnati, "Wireless sensor networks", kluwer academic publishers.	2004
6	Feng Zhao and Leonidas J. Guibas, "Wireless Sensor Networks:An Information Processing Approach", Elsevier.	2004
7	Robert Faludi, "Building Wireless Sensor Networks", O'Reilly.	2010
8	Holger Karl and Andreas Willig, "Protocols and Architectures for Wireless Sensor Networks", John wiley and sons.	2005
9	Rajeev Shorey, A. Ananda, MunChoon Chan and Wei Tsang Ooi, "Mobile, wireless, and Sensor networks - technology, applications, and future directions", IEEE press and wiley interscience.	2003

Course Name	**:**	**MOBILE COMPUTING**
Course Code	**:**	**CSN 418**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:

The students should be able to understand the concepts and principles of mobile computing. They should explore both theoretical and practical issues of mobile computing. They should develop skills of finding solutions for mobile computing applications

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**UNDERSTAND MOBILE PHONES** Mobile Phones -- The New Platform for People-Centric Sensing, Social and Personal Applications , Localization, Exploiting Mobility , Location Privacy	**8**
2	**MOBILE TECHNOLOGIES** Anatomy of a Mobile Device, Survey of Mobile Devices, Applications of Mobile Computing	**4**
3	**WIRELESS COMMUNICATION TECHNOLOGIES** Cellular networks, Wireless (802.11), TCP/IP in the mobile setting, Geolocation and GPS	**6**
4	**APPLICATION DESIGN** Context, Information Architecture, Design Elements, Mobile Web vs Native Applications	**6**
5	**SENSORS ON MOBILE PHONES** Function of sensors: Accelerometer, GPS, Gyroscope, Magnetometer, Luxmeter, Microphone, Proximity Sensor	**6**
6	**DEVELOPMENT ENVIRONMENTS** Introduction to Objective-C, The Model-View-Controller Model, The Delegate Pattern, The iPhone, Android, & Blackberry SDKs, The Application Environment, Limited Resource Computing, Memory Management, Low Power Computing, Fault Tolerance and Persistence, Security Issues	**12**

List of Experiments:		**Number of Turns**
1	Getting familiar with Mobile OS (Android platform considered here) and its Integrated Development Environment. Create single screen/activity application demonstrating the use of: Layouts, Widgets, and Menu components to compose the user interface: a. Implement a program demonstrating various possibilities for changes in the color, size and font style (of the displayed TextView content), as a result of pressing different buttons or selecting various menu options. b. Create and implement a general-purpose mathematical calculator. c. Develop some kind of multiple-choice e-learning application, that supports memorizing the correlation of predefined number of word-pairs, in two different languages (e.g. Polish and English)	04
2	• Mobile Apps be developed for different application for Android as well as IoS Platform. Theme of final Android project may be proposed by student. • Each student selects at least two of mobile technology aspects: a. mobile database (SQLite) b. built-in smartphone sensors: accelerometer, gyroscope, magnetometer, gps c. networking, data synchronization or web services d. 3D graphics or animation on mobile device e. Bluetooth communication between mobile devices	04
3	Design Loggers to perform User Localization Through Sensors of Smartphones. Collect data using the designed mobile App and present the results.	03
4	Develop Mobile Apps based upon Crowdsourcing-Data Collection and Data Analysis	02

Course Outcomes: At the end of the course, students will be able to:	
1	Grasp the concepts and features of mobile computing technologies and applications
2	Identify the important issues of developing mobile computing systems and applications
3	Organize the functionalities and components of mobile computing systems into different layers and apply various techniques for realizing the functionalities

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Theodore S. Rappaport, Wireless Communications: Principles and Practice, Second Edition, Prentice Hall,	2002
2	Ivan Stojmenovic, Handbook of Wireless Networks and Mobile Computing, John Wiley & Sons	2002
3	Mohd. Ilyas & Imad Mahgoub, Mobile Computing Handbook, CRC Press/Aurbach Publications, ISBN 0-8493-1971-4, Boca Raton USA.	2005
4	Reza B'Far, Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML, Cambridge University, 2004	2004
5	Reto Meie, Professional Android Application Development (Wrox Programmer to Programmer), Wrox, 2008	2008
6	Axel Küpper, Location-Based Services: Fundamentals and Operation, Wiley, 2005.	2005

Course Name	**:**	**OBJECT ORIENTED PROGRAMMING**
Course Code	**:**	**CSN461**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
The students should be able to understand the concept of object oriented programming like classes, constructors, polymorphism, inheritance, templates and file handling.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION TO PROGRAMMING PARADIGMS** Introduction to various programming paradigms, advantages of OOP, comparison of OOP with Procedural Paradigm	**2**
2	**C++ CONSTRUCTS** Tokens, Expressions and control structures, various data types and data structures, Variable declarations, Dynamic Initializations, Operators and Scope of Operators, Typecasting, Unformatted and formatted console I/O Operations	**2**
3	**FUNCTIONS, CLASSES AND OBJECTS** Prototyping, Referencing the variables in functions, Inline, static and friend functions. Memory allocation for classes and objects. Arrays of objects, pointers to member functions.	**4**
4	**CONSTRUCTORS AND DESTRUCTORS** Constructor and Destructor types, Dynamic Constructors, Applications, Order of Invocation, C++ garbage collection, dynamic memory allocation.	**5**
5	**POLYMORPHISM** Function and Operator overloading, overloading using friend Functions, type conversions from basic data types to user defined and vice versa.	**5**
6	**INHERITANCE** Base classes and Derived classes, types of inheritance, various types of classes, Invocation of Constructors and Destructors in Inheritance, aggregation, composition, classification hierarchies, metaclass/abstract classes.	**5**
7	**POINTERS** Constant pointers, Use of this Pointer, Pointer to derived and base classes, virtual functions, Bindings, Pure virtual Functions and polymorphism.	**5**
8	**I/O OPERATIONS AND FILES**	**5**

	Classes of files, Operations on file, file pointers.	
9	**GENERIC PROGRAMMING WITH TEMPLATES** Definition of class template, Function Templates, Overloading Template Functions, Class templates and member functions templates with parameters, Standard C++ classes, persistent objects, streams and files, namespaces, exception handling, generic classes, standard template library: Library organization and containers, standard containers, algorithm and Function objects, iterators and allocators, strings, streams, manipulators, user defined manipulators and vectors.	**4**
10	**ADVANCED MEMORY HANDLING** Storage types in C++, Automatic Life, Dynamic Life, Static Life, Object with special storage restrictions.	**2**
11	**CASE STUDY** Features of Different Object Oriented languages	**3**

List of Experiments:		**Number of Turns**
1	Implement various C++ constructs	**1**
2	Implement friend functions	**1**
3	Implement pointers to member functions	**2**
4	Implement constructors and destructors	**1**
5	Implement operator and function overloading	**2**
6	Implement inheritance	**1**
7	Implement run time polymorphism	**2**
8	Implement file operations	**2**
9	Implement templates	**1**

Course Outcomes: At the end of the course, students will be able to:	
1	Understand real world problem and identify object in given problem.
2	Construct C++ classes and apply various C++ concepts with proficiency

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	C++ Primer, 5th Edition, Pearson Education Inc.	2012
2	Object Oriented Programming in C++, Robert Lafore, 4th Edition, SAMS	2001
3	C++ Primer Plus By Prata, Pearson Education	2012
4	C++:The Complete Reference, By Schildt, McGraw-Hill	2003
5	Object Oriented Programming with C++, Balaguruswamy, Tata Mc Graw Hill	2008

Course Name	**:**	**OPERATING SYSTEMS**
Course Code	**:**	**CSN462**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
Students should be able to describe the services provided by and the design of an operating system. They should be able to understand the structure and organization of the file system, processes synchronization, process scheduling, system calls and different approaches to memory management.

Total No. of Lectures – 42

Lecture wise breakup		Number of Lectures
1	**CONCEPTS OF OPERATING SYSTEMS** Computer system overview, concept of an operating system, batch system, multiprogramming, multiprocessing, multi user, time sharing, personal system, parallel system, real time system, simple monitors, general system architecture, System components, operating system services, system calls, system programs, system structure, Approaches to OS design and implementation: Microkernel, Layered, Kernel Approach	**8**
2	**PROCESSES AND THREADS** Concept of process, process states, process state transitions, process control block, operations on processes, threads, concurrent processes, mutual exclusion and synchronization, principles of deadlocks, integrated deadlocks strategy, scheduling levels, scheduling criteria, Inter process synchronization, Inter process communication, Linux, IPC Mechanism, Remote procedure calls, RPC exception handling, security issues	**8**
3	**MEMORY MANAGEMENT** Logical and physical address space, storage allocation and management techniques, swapping concepts of multi programming, paging, segmentation, virtual storage management strategies, demand paging, page replacement algorithm, thrashing	**8**
4	**INPUT/OUTPUT AND DATA MANAGEMENT** File organization, record blocking, access method, directory structure, protection file system structure, allocation methods, free space management, directory implementation, disk structure, disk scheduling, disk management, buffering, swap space management, RAID levels	**8**
5	**OS SECURITY** Types of Threats in OS, Basic OS Security Mechanisms, Understanding the Threats: Malware Taxonomy: Viruses, Worms, Rootkits, Defense -- An Overview, Logging, Auditing, and Recovery, OS-level Memory Protection	**4**
6	**CASE STUDIES** Linux/Unix OS design and architecture, Unix shell, Unix operating system services, user perspective, representation of files in Unix system processes and their structure, input-output system, memory management in Unix	**4**
7	**OS ABSTRACTIONS** Processes: fork, wait, exec, exit, kill, getpid, brk, nice, sleep, trace Files: open, close, read, write, lseek, stat, sync Directories: mkdir, rmdir, link, unlink, mount, umount users + Security: chown, chmod, getuid, setuid, Inter process communication: signals, pipe Networking: socket, accept, snd, recv, connect	**2**

List of Experiments:		Number of Turns
1	To perform shell programming	2
2	Implement memory management techniques like paging or segmentation	2
3	Implement any file allocation technique (Linked, Indexed or Contiguous)	2
4	Use the system calls of UNIX operating system: mkdir, rmdir, link, unlink, mount, umount users +	2
5	Use the following system calls of UNIX operating system: fork, wait, exec, exit, kill, getpid, brk, nice, sleep, trace, open, close, read, write, lseek, stat, sync	2
6	Use the following system calls of UNIX operating system: mkdir, rmdir, link, unlink, mount, umount users +, chown, chmod, getuid, setuid	2
7	Use the following system calls of UNIX operating system: signals, pipe, socket, accept, snd, recv, connect	1

Course Outcomes: At the end of the course, students will be able to:	
1	Explain about operating systems and its major components.

2	Write and/or modify concurrent programs.
3	Apply security as well as recovery feature in the design of algorithms.

Suggested Books:		
Sr. No.	**Name of Book/ Authors/ Publisher**	**Year of Publication/ Reprint**
1	Operating system, Galvin & Silberschatz, 7th Edition, John Willey	2004
2	Operating Systems-A Concept Based Approach By Dhamdhare, TMH	2006
3	Operating systems Internals and design principles By William Stallings, Pearson Education	2012
4	Operating Systems –A Design Oriented approach By Crowley, TMH	2001
5	Operating systems Design and Implementation By Andrew S. Tanenbaum, Pearson Education	2009

Course Name	**:**	**WEB TECHNOLOGIES**
Course Code	**:**	**CSN463**
Credits	**:**	**4**
L T P	**:**	**3 0 2**

Course Objectives:
Students should understand fundamentals of programming in Java. They should be able to design and develop Java programs using JDK environment. They should learn the basics about Client side scripts and Server side scripts.

Total No. of Lectures – 42

Lecture wise breakup		**Number of Lectures**
1	**INTRODUCTION TO JAVA** Creation of Java, importance to Internet, Java applets and applications, security, portability, Java's Byte code. Understanding the java programs, compiling the program, control statements, lexical issues, Data types in Java, Java literals, variables, scope and lifetime of variables, type conversion, declaring and using arrays	4
2	**OPERATORS IN JAVA** Arithmetic, Modulus, Assignment, Bitwise, Relational, Assignment, ternary operator, Operator precedence. Control statements, classes, garbage Collection, overloading methods, overloading constructors, using objects as parameters. Argument passing by value and by reference, returning objects, static and final keywords. Inheritance in Java using super keyword, abstract classes.	6
3	**EXCEPTION HANDLING IN JAVA** Exception types, using try and catch, multiple catch classes, nested try statement, using throw, Java Built in Exceptions, creating user defined exceptions.	6
4	**MULTITHREADED PROGRAMMING IN JAVA** Java thread model, Thread priorities, synchronization, creating thread using thread class and runnable interface, creating multiple threads, deadlocks.	6
5	**APPLETS AND AWT** Applet fundamentals, Applet architecture, creating and running applets, Passing parameters in Applets. AWT : AWT Basics, AWT classes, Working with event frame windows, working with graphics, creating and selecting a font, Managing Text output using Font Metrics.	6
6	**USING AWT CONTROLS** Menus, using Labels, buttons, checkboxes, Checkbox group, choice controls, using Lists, managing scrollbars, using a textField, textArea.	4