SCHEME AND SYLLABUS FOR PROGRAM

B.TECH COMPUTER ENGINEERING

w.e.f. 2017-18

SCHEME

B.Tech. Computer Engineering I Semester w.e.f 2017-18

Course Code	Course Title	Prerequisites Course Code(s)	L	T	P	Credits	Course Type
HSIR11	Communication Skills in English		2	0	2	3	EPR
MAIR11	Mathematics-I		3	1	0	4	EPR
PHIR11	Physics-I		2	1	2	4	EPR
CHIR11	Environment Studies		2	0	2	3	EPR
CSIR11	Basics of Programming		2	0	2	3	EPR
CSIR13	Introduction to Discipline Engineering		2	0	0	2	EPR
MEIR11	Engineering Practice		1	0	3	2	EPR
SWIR11	Sports		0	0	2	1	EPR
		Total	**14**	**2**	**13**	**22**	

B.Tech. Computer Engineering II Semester w.e.f 2017-18

Course Code	Course Title	Prerequisites Course Code(s)	L	T	P	Credits	Course Type
HSIR12	Economics for Engineers		2	1	0	3	EPR
MAIR12	Mathematics-II		3	1	0	4	EPR
PHIR12	Physics-II		2	1	2	4	EPR
CHIR12	Chemistry		2	1	2	4	EPR
CSPC10	Digital System Design		3	0	0	3	PC1
CSPC12	Data Structures		3	0	2	4	PC2
CSPC14	Object Oriented Programming using C++		3	0	2	4	PC3
CSIR12	Engineering Graphics (Web Designing)		1	0	3	2	EPR
		Total	**19**	**5**	**11**	**28**	

B.Tech. Computer Engineering III Semester w.e.f 2017-18

Course Code	Course Title	Prerequisites Course Code(s)	L	T	P	Credits	Course Type
MAIR24	Discrete Mathematics		3	1	0	4	EPR
CSPC21	Design and Analysis of Algorithms		3	1	2	5	PC1
CSPC23	Java Programming		3	0	2	4	PC2
CSPC25	Database Systems		3	1	2	5	PC3
CSPC27	Software Engineering		3	1	0	4	PC4
CSPC29	Computer Organization and Architecture		3	1	0	4	PC5
		Total	**18**	**5**	**8**	**26**	

B.Tech. Computer Engineering IV Semester w.e.f 2017-18

Course Code	Course Title	Prerequisites Course Code(s)	L	T	P	Credits	Course Type
CSPC20	Operating Systems		3	1	2	5	PC1
CSPC22	Microprocessor and Interfacing		3	0	2	4	PC2
CSPC24	Unix and Linux Programming		2	0	2	3	PC3
CSPC26	Computer Networks		3	1	2	5	PC4
CSPE2x	Programme Elective I		3	0	2	4	PE1
CSPE2x	Programme Elective II		3	1	0	4	PE2
		Total	**17**	**3**	**10**	**25**	

List of Programme Electives I		**List of Programme Electives II**	
Course Code	**Course Title**	**Course Code**	**Course Title**
CSPE20	Advanced Java Programming	CSPE60	Stochastic Processes and Queuing Theory
CSPE22	Programming using Python	CSPE62	Software Project Management
CSPE24	Number Theory and Cryptography	CSPE64	Data Mining and Data Warehousing
CSPE26	Efficient Coding for Problem Solving and Competitive Programming		

B.Tech. Computer Engineering V Semester w.e.f 2017-18

Course Code	Course Title	Prerequisites Course Code(s)	L	T	P	Credits	Course Type
CSPC31	Advanced Data Structures and Algorithms		3	0	2	4	PC1
CSPC33	Soft Computing		3	0	2	4	PC2
CSPC35	Software Development using UML and Agile Methodology		3	0	2	4	PC3
CSPC37	Automata Theory		3	1	0	4	PC4
CSPC39	Seminar		0	0	2	1	
CSPE3x	Programme Elective III		3	0	2	4	PE1
CSPE3x	Programme Elective IV		3	1	0	4	PE2
		Total	**18**	**2**	**10**	**25**	

List of Programme Electives III		**List of Programme Electives IV**	
Course Code	**Course Title**	**Course Code**	**Course Title**
CSPE31	Software Verification and Validation	CSPE71	Advanced Database Systems
CSPE33	Embedded System Development	CSPE73	Service Oriented Architecture
CSPE35	Mobile Application Development	CSPE75	Advanced Computer Architecture
CSPE37	Complex Networks	CSPE77	Image Processing

B.Tech. Computer Engineering VI Semester w.e.f 2017-18

Semester VI							
Course Code	**Course Title**	**Prerequisites Course Code(s)**	**L**	**T**	**P**	**Credits**	**Course Type**
CSIR30 / CSIR32	Internship / Project Work					10	EPR
					VI Semester Credits	**10**	

B.Tech. Computer Engineering VII Semester w.e.f 2017-18

Course Code	Course Title	Prerequisites Course Code(s)	L	T	P	Credits	Course Type
HSIR13	Business Management		2	1	0	3	EPR
CSPC41	Machine Learning		3	0	2	4	PC1
CSPC43	Distributed Computing		3	0	0	3	PC2
CSPC45	Information Security		3	0	2	4	PC3
CSPE4x	Programme Elective V		3	1	0	4	PE1
	Open Elective I from other departments		3	0	0	3	OE1
		Total	**17**	**2**	**4**	**21**	

List of Programme Electives V		**Open Elective I (For Non-CS and IT students)**	
Course Code	**Course Title**	**Course Code**	**Course Title**
CSPE41	Cloud and Grid Computing	CSOE41	Operating Systems
CSPE43	Natural Language Processing		
CSPE45	Computer Vision		
CSPE47	Graphics and Visualization		

B.Tech. Computer Engineering VIII Semester w.e.f 2017-18

Course Code	Course Title	Prerequisites Course Code(s)	L	T	P	Credits	Course Type
HSIR14	Professional Ethics & IPR		1	0	2	2	EPR
CSPC40	Big Data Analytics		3	0	2	4	PC1
CSPE4x	Programme Elective VI		3	1	0	4	PE1
CSPE40	PROJECT		0	0	3	4	PE2
	Open Elective II from other departments		3	0	0	3	OE2
	Open Elective III from other departments		3	0	0	3	OE3
CSIR40	Comprehensive Viva Voce					3	
		Total	**13**	**1**	**7**	**23**	

List of Programme Electives VI	
Course Code	**Course Title**
CSPE42	High Performance Computing
CSPE44	Human Computer Interaction
CSPE46	Compiler Design
CSPE48	Graph Theory and Combinatorics
Open Elective II and III	

Course Code	Course Title
CSOE40	Database System
CSOE42	Soft Computing

Course Code	**:**	**CSIR11**
Course Title	**:**	**Basics of Programming**
Number of Credits	**:**	**3**
Prerequisites	**:**	
Course Type	**:**	**EPR**

Course Learning Objectives:

The objective of this course is to provide fundamentals of problem solving using C language programming.

Course content:

1. **Programming Fundamentals & Control Statements:**
   Block Diagram of Computer, Hardware vs software, concept of operating system and compiler, software development life cycle and structured programming. Introduction to C programming, basic programming using input and output operators and expressions, programming using if and if-else, Programming using looping-for, while, do-while; use of switch and break.

2. **Arrays based Programming**
   Defining and processing 1-d and 2-d arrays for problem solving.

3. **Modular programming using Functions**
   Defining and calling a function, modular programming using functions, passing arguments and arrays to functions, functions of void and returning values.

4. **Programming using Strings & Structures**
   Defining and processing string as array of character, use of null char, defining and processing structures, passing strings and structures to functions.

BOOKS

1. Byron S. Gottfried, Programming with C Language, Schaum Series, Tata McGraw Hill, 2015.
2. E Balaguruswamy, Programming with C, Tata McGraw Hill, 2015.
3. Kernighan & Richie, C Programming, Prentice Hall of India, 2002.

Course Outcomes:

1. Understand the use of software and programming for problem solving.
2. Learn programming using simple concepts of input, output and control statements.
3. Use arrays, functions, strings and structure for problem solving.

Course Code	**:**	**CSIR13**
Course Title	**:**	**Introduction to Discipline Engineering**
Number of Credits	**:**	**2**
Prerequisites	**:**	
Course Type	**:**	**EPR**

Course Learning Objectives:

The objective of this course is to provide knowledge of the core branch of Engineering and Technology and different courses that the student is going to study in next four years.

Course content:

1. Overview of CSE & IT as a discipline, differentiation between CS & IT, Broader industrial applications of CSE &IT such as software development, Database oriented applications, computer networks and related applications, System oriented scope at operating system level and hardware level.

2. Programming oriented directions and scope, competitive programming, algorithms and their performance related issues; Necessity of data base systems and its applications, discussion about data mining; Conceptual overview of Computer Architecture, operating system's functioning, automata theory and its applications, Basics of Computer Networks and recent developments in this area; Visit to department labs, CCN and server rooms, library and Institute automation software.

3. Overview of adding intelligence to machines using soft computing, Importance of Information security and introduction to basis terminologies such as virus, worm, Trojan, vulnerabilities, denial of services, ethical hacking Overview of Applications and scope of mobile application development, image processing and natural language processing.

4. Recent Developments in the area of Computer Vision, and Human Computer Interaction Awareness about cloud and distributed computing Technologies and issues behind Social networking oriented system design and development Futuristic areas like machine learning, big data analytics.

Course Outcomes:

1. Student will be aware of different domains of the computer engineering and information technology.
2. Student will be able to distinguish between core and elective courses being offered in the programme of study.
3. Student will become aware of facilities and infrastructure of the department.

Course Code	**:**	**CSPC10**
Course Title	**:**	**Digital System Design**
Number of Credits	**:**	**3**
Prerequisites	**:**	
Course Type	**:**	**PC**

Course Learning Objectives:

1. To provide the basic knowledge gates, Number Systems and their Arithmetic, Computer Codes, digital logic families, SOP and POS simplifications.
2. Ability to design MSI circuits.
3. Gain Knowledge about the function of Flip-Flops, Counters, Registers.
4. Understanding the concepts of Sequential circuits.

Course Content

1. **Number Systems and Codes**
   Introduction to the positional number system, signed magnitude numbers, floating point numbers, binary arithmetic: addition, subtraction, multiplication and division, Base conversion, conversion formulas with examples, one's and two's compliment arithmetic, Computer codes – BCD codes, gray codes, excess-3 codes, parity checks, Hamming and alphanumeric codes.

2. **Digital Logic Families**
   Introduction to digital ICs, TTL, Schottky TTL, ECL, MOS Logic, CMOS Logic, Tri-state logic: Characteristics and properties.

3. **Combinational Logic Design**
   Introduction, standard representations for logical functions, Karnaugh map representation, simplification of logical functions using K-map, minimization of logical functions specified in minterms/maxterms or Truth Table, minimization of logical functions not specified in minterms/maxterms, Don't care conditions, design examples, Ex-or and Ex-nor simplification of K-

maps, five and six-variable K-maps, QM method, MEV method, Introduction of multiplexers and their use in combinational logic design, demultiplexers /decoders and their use in combinational logic design, adders and their use as subtractors, digital comparators, parity generators/checkers, code converters, priority encoders, 7-segment decoder/driver.

4. **Synchronous Sequential Circuits and Asynchronous Sequential Circuits**
   Introduction, FSM model, memory elements and their excitation functions. Synthesis of synchronous sequential circuits, capabilities and limitation of FSM, state equivalence and minimization, simplification of incompletely specified machines, Fundamental mode circuits synthesis, state assignment, pulse mode circuits.

Reference Books:

1. R.P. Jain: Modern Digital Electronics, Tmh.
2. Z Kohavi: Switching and Finite Automata Theory, Tmh
3. M.M. Mano: Digital Logic Design, Phi.
4. Dr. B.R. Gupta: Digital Electronics, Katson.
5. James W. Bignell& Robert Donovan: Digital Electronics, Cengage Learning.
6. Sanjay Kumar Bose: Digital Systems, New Age International Publishers

Course outcomes:

1. Understand the basic fundamentals of digital systems.
2. Apply fundamental to implement real systems that are useful for the society.
3. Understand the basic concept of synthesis of synchronous as well as asynchronous sequential circuits.

Course Code	**:**	**CSPC12**
Course Title	**:**	**Data Structures**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSIR11**
Course Type	**:**	**PC**

Course Learning Objectives

1. This course introduces the concept of Data Structures used in various computer science applications.
2. The students are introduced to understand and efficiently apply various data structures such as stacks, queues, linked lists, trees and graphs for solving various computing problems using C programming language.

Course Content

1. **Fundamentals of pointers & Files:**
   Pointers in C: pointer declaration, passing pointer to functions, pointers and arrays, dynamic memory allocation, Files and related operations. Searching techniques: Linear and Binary, Sorting techniques: Selection, Bubble, Insertion, Merge sort, Quicksort.

2. **Simple Data Structures**
   Arrays based Linear Data Str: Array storage, sparse arrays; Transpose, addition, and multiplication of sparse matrices, Stacks and Queues and their applications, multiple stacks and queues in an array.

3. **Linked Data Structures**
   Singly, Doubly & Circular Linked Lists; representation, operations and applications, linked stacks and queues. Linked lists based polynomial addition

4. Advanced Data Structures

Trees, Basic concepts and definitions of a tree and binary tree and associated terminology, Binary tree traversal techniques, some more operations on binary trees, Heaps and heapsort.

Reference Books

1. E Horowitz and S. Sahni: Fundamentals of Data Structures in C, Second Edition, Universities Press, Hyderabad.
2. R.L. Kruse: Data Structures & Program Design in C, PHI.
3. D.F. Knuth: The art of Computer Programming Vol 1, Narosa Publications, 1985.
4. Byron S. Gottfried & J K Chhabra: Theory and Problems of Programming with C Language, Schaum's Outlines Series, TMH, 2005.

Course Outcomes

1. Develop skill to identify and determine the usage of various data structures, operations, associated algorithms and implement their applications.
2. Apply knowledge of pointers, memory allocation and string handling for solving programming problems.
3. Understand the concept of trees and graphs, their implementation and applications.
4. Able to implement standard algorithms for searching and sorting.
5. Analyze efficiency of different algorithms using time and space complexity.

Course Code	**:**	**CSPC14**
Course Title	**:**	**Object Oriented Programming using C++**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSIR11**
Course Type	**:**	**PC**

Course Learning Objectives:

1. Able to differentiate between structure oriented programming and object oriented programming and to specify simple abstract data types and design implementations
2. Recognize features of object-oriented design such as encapsulation, polymorphism, inheritance and composition of systems based on object identity.
3. Able to use object oriented programming language like C++ and associated library to develop object-oriented programs.
4. Able to understand and to use basic fundamentals of Java.

Course Content:

1. **Object Oriented Programming and Design**

   Review of Abstraction, Objects and other basics, Encapsulation, Information hiding, Method, Signature, Classes and Instances, Polymorphism, Inheritance, Exceptions and Exception Handling with reference to object modeling, Object Oriented Design – Process, Exploration and Analysis.

2. **C++ Programming Basics**

   Fundamentals: Variables and assignments, Input and Output, Data Types and Expressions, Flow of control, Subprograms: Top down design, Predefined functions, Programmer defined functions, Procedural abstractions, Local variables, Overloading function names, Parameter passing, Structures for diverse data, Structures as function arguments, Initializing structures, Defining classes and member functions, Public and private members ,Constructors for initializations, , Destructors, Copy constructors, Friend functions.

3. **C++ Object Oriented Concepts**

   Objects and Classes: Operator overloading , Overloading the assignment operator ,subscript operator , this pointer ,Use of file for I/O, Formatting output with stream functions, Character I/O, Inheritance, constructors in inheritance , Standard C++ classes, Derived classes ,Virtual functions, virtual base class.

4. **C++ Data Structures and Advanced Topics**

Arrays – Programming with arrays, arrays of classes, arrays as function arguments, Strings, Multidimensional arrays, Arrays of strings, Pointers, Dynamic arrays, Classes and dynamic arrays, Templates – generic classes and functions, namespaces, introduction to STL.

5. **Introduction to Java**
   Data types, Variables and Assignment, String and Characters, Arrays, Control statements, Loops, Operators. Introduction to Classes, Constructors, this keyword, Static, Local and Instance variables, Methods, Method overloading, Method overriding, subclasses, inheritance, modifiers, polymorphism.

BOOKS

1. Herb Schildt: C++ - The Complete Reference, TMH, Delhi
2. R.Venugopal :Mastering C++,TMH, Delhi
3. Bruce Eckel : Thinking in C++ Volume I ,Pearson Education, Delhi.
4. Horstmann: Computing Concepts with C++ Essentials, John Wiley.
5. Joseph O'Neil and Herb Schildt: Teach Yourself JAVA, TMH, Delhi.
6. Gary Cornell, Sun Microsystems: Core Java 2,Pearson Education, Delhi.

Course Outcomes

1. Able to differentiate between structure oriented programming and object oriented programming and to specify simple abstract data types and design implementations
2. Recognize features of object-oriented design such as encapsulation, polymorphism, inheritance and composition of systems based on object identity.
3. Able to use object oriented programming language like C++ and associated library to develop object-oriented programs.
4. Able to understand and to use basic fundamentals of Java.

Course Code	**:**	**CSIR12**
Course Title	**:**	**Engineering Graphics (Web Designing)**
Number of Credits	**:**	**2**
Prerequisites	**:**	
Course Type	**:**	**EPR**

Course Learning Objectives:

1. Introduction and brief history of World Wide Web (WWW).
2. Web essentials: HTML, XHTML, CSS.
3. Addressing web standards, audience requirements and principles of web page design.
4. Introduction of Web architecture.

Course Content:

1. **Introduction:** Brief history of internet, introduction to world wide web, basic principles involved in developing a web site, rules of web designing, web standards, audience requirements, Design concept.

2. **Web essentials and standards**: Clients, servers, introduction to Markup languages, scripting languages, Introduction to elements of HTML, XHTML and CSS, Introduction to Document object model (DOM), working with text, list, tables, frames, hyperlinks, Images multimedia, forms and controls. CSS properties, Id and Class, Box Model, creating page Layout and Site Designs

3. **Javascript** Javascript as programming language, Data types, Values, Variables, Expressions and Operators. JavaScript Statements, loops, arrays, strings, methods, Defining and Invoking functions and their closure, random functions and maths library, representing dates. Pattern Matching and Regular Expressions. Javascript in web browsers, difference between server side and client side javascript, embedding javascript in HTML and frameworks, Changing CSS style, hiding HTML elements, showing hidden HTML elements. DOM and event handling, error handling, mouse, text,

drag, drop and keyboard events and node operations, Node operations, Cookies, Scripted HTTP, Animation and multimedia Forms of Debugging.

4. **Website Development Tools** Google Web Designer, Macaw, Sketch, Firefox, YSlow, Wordpress, openElement etc.

Reference Books:

1. Thomas A Powell, HTML: The Complete Reference, Tata McGraw Hill Publications.
2. Scott Guelich, Shishir Gundavaram, Gunther Birzniek; CGI Programming with Perl 2/e, O'Reilly
3. Doug Tidwell, James Snell, Pavel Kulchenko; Programming Web Services with SOAP, O' Reilly
4. Robert. W. Sebesta, "Programming the World Wide Web", Fourth Edition, Pearson Education, 2007.
5. Yong, XML Step by Step, PHI.
6. Chris Bales, "Web programming- Building Internet Application".
7. Deitel, Deitel, Goldberg, "Internet & World Wide Web How To Program", Third Edition, Pearson Education, 2006.
8. Marty Hall and Larry Brown, "Core Web Programming" Second Edition, Volume I and II, Pearson Education, 2001.

Course outcomes

1. Understand basic principles of web site design, considering the information architecture.
2. Incorporate best practices in navigation, usability in website design
3. Design of website adhering to current web standards (HTML, XML, CSS)
4. Learning various scripting languages to create interactive components in web pages.

Course Code	**:**	**CSPC21**
Course Title	**:**	**Design and Analysis of Algorithms**
Number of Credits	**:**	**5**
Prerequisites	**:**	**CSPC12**
Course Type	**:**	**PC**

Course Learning Objectives:

1. Able to design, implement and analysis of standard searching and sorting algorithms.
2. Implement standard divide and conquer, Dynamic programming, Greedy and backtracking algorithms.
3. Able to implement between different data structures i.e., trees, heaps etc. also, able to pick an appropriate data structure for any given design situation.
4. Able to implement the major graph algorithms and their analysis.

Course Content:

1. **Introduction**
   Concept of Time and space complexity, analysis of algorithms, asymptotic notation, recurrence relations, design and analysis of D & C problems like quick sort etc, heap sort, priority queues, sorting in linear time, hashing, binary search trees.

2. **Graph Algorithms**
   Graph representation & traversal (search), topological sort, strongly connected components, minimum spanning trees – Kruskal and Prim's, Single source shortest paths, relaxation, Dijkstra's algorithm, Bellman-Ford algorithm, single source shortest paths for directed acyclic graphs, all-pairs shortest path.

3. **B-Trees and Dynamic programming**
   B-Trees: representation and operations; Elements of Dynamic Programming, structure and steps, Matrix-chain multiplication, longest common subsequence.

4. **Greedy & Backtracking Approaches:**

Greedy algorithms – Elements, activity-selection problem, Huffman codes, task scheduling problem, Knapsack Problem, Backtracking – Elements, 8 – Queens, Graph Coloring, Hamiltonian Cycles.

BOOKS

1. Cormen, Leiserson and Rivest: Introduction to Algorithms, 3/e, PHI.
2. Horowitz, Sahni, and Rajasekaran: Fundamentals of Computer Algorithms, Second Edition, Universities Press, Hyderabad.
3. Aho, Hopcroft, and Ullman: The Design and Analysis of Computer Algorithms, Addison Wesley.

Course Outcomes (CO's)

1. Able to design, implement and analysis of standard searching and sorting algorithms.
2. Implement standard divide and conquer, Dynamic programming, Greedy and backtracking algorithms.
3. Able to implement between different data structures i.e., trees, heaps etc. also, able to pick an appropriate data structure for any given design situation.
4. Able to implement the major graph algorithms and their analysis.

Course Code	**:**	**CSPC23**
Course Title	**:**	**Java Programming**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSIR11**
Course Type	**:**	**PC**

Course Learning Objectives:

1. Building robust applications using Java's object-oriented features.
2. Understanding the usage of java class libraries.
3. Building multithreaded, platform-independent and GUI based java applications for business problems.

Course Content:

1. The overview of Java's architecture and the architecture of the Java Virtual Machine (JVM).
Classes: Declaring Members (Fields and Methods), Instance Members, Static Members.
Objects: Class Instantiation, Reference Values, and References, Object Aliases. Basic Language Elements, Primitive Data Types, Variable Declarations, Initial Values for Variables, Class Declarations, Method Declarations, this reference, Method Overloading, Constructors, The Default Constructor and Constructors overloading. Arrays, Anonymous Arrays, Multidimensional Arrays, Variable Arity Methods, The main() Method, Program Arguments.

2. **Packages:** Defining Packages, Using Packages, Compiling Code into Packages, Running Code from Packages. Scope Rules, Accessibility Modifiers, Overview of other Modifiers for Members. Operators and Expressions**,** Overview of Control Flow Statements.
Exception Handling: The try Block, The catch Block, The finally Block, The throw Statement, The throws Clause, Checked and Unchecked Exceptions, Defining New Exceptions.

3. **Object-Oriented Programming:** Single Implementation Inheritance, Overriding Methods, Hiding Members, The Object Reference super, Chaining Constructors Using this() and super()
Interfaces: Defining Interfaces, Abstract Method Declarations, Implementing Interfaces, Extending Interfaces, Interface References, Constants in Interfaces, Polymorphism and Dynamic Method Lookup.
Fundamental Classes: Overview of the java.lang Package, The Object Class, The Wrapper Classes, The String Class, The StringBuilder and the StringBuffer Classes.

4. **Multithreading:** Overview of Threads, the Main Thread, Thread Creation, Synchronization, Thread Transitions. Basics of Event Handling, Graphics Programming using AWT and Swing, An overview of the new Features of Java 7 & 8.

Reference Books:

1. Bruce Eckel, Thinking In Java, Pearson Education, 4th Ed., 2006.
2. Dietel & Deitel, Java How to Program, Pearson Education, 10th Ed., 2015.
3. Kathy Sierra & Bert Bates, Head First Java, O'REILLY, 2nd Ed., 2005.
4. Cay s. Horstmann & Gary Cornell, Core Java. Volume I, Fundamentals, Sun Microsystems Press, 8th Ed., 2008.

Course outcomes:

1. Write Java programs that solve simple business problems.
2. Create java applications that are robust and multithreaded.
3. Write simple GUI interfaces for a program to interact with users, and to understand the event-based GUI handling principles.

Course Code	**:**	**CSPC25**
Course Title	**:**	**Database Systems**
Number of Credits	**:**	**5**
Prerequisites	**:**	
Course Type	**:**	**PC**

Course Learning Objectives:

1. To understand fundamentals of data models and to conceptualize a database system for user requirement.
2. To study the fundamental of Database query language, like SQL and relational algebra.
3. To learn the concept of normalization in database design.
4. To learn fundamental concepts of transaction processing, concurrency control techniques and database recovery procedure.
5. Understand the professional, ethical and security issues and responsibilities in database design.

Course Content:

1. **Introduction:** Overview of file systems and Database Systems , Software Architecture of a typical DBMS, Data Models, Relational Data Model, Network Data Model, Hierarchical Data Model, Schemas and Instances, Database Administration and Control.

2. **Relational Model and Query Language**: Overview of Relational Database, Key Integrity constraint, Relational Algebra, Relational Calculus: Domain Relational Calculus, Tuple Relational Calculus, SQL Fundamentals, Basic operators, Missing Information, Null Values, Additional Operations, Advanced SQL features, Embedded SQL, Dynamic SQL, Database Views.

3. **Database Design**: Overview of Normalization, Database Anomalies, Functional Dependencies, Candidate and Super Key, Non-loss Decomposition, Dependency Preservation, Normal forms: First, Second, Third Normal, Boyce Codd Normal Form, Multi-valued Functional Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form, Denormalization.

4. **Transaction Processing**: Overview of Database Transactions, Transaction states, Transaction Recovery, ACID Properties, Transaction Recovery: Two Phase Commit, Commit Points, Serializability, Concurrency control: Need for Concurrency, Locking Protocols: Binary lock, Two Phase Lock, Deadlock, Starvation, Transaction Timestamp.

Reference Books

1. A Silberschatz, H.F. Korth & S. Sudarshan: Data Base System Concepts, TMH, 1997.
2. A.K. Majumdar and Bhattacharyya: Database Management Systems, THM, 1996.
3. C.J. Date: An Introduction to Database systems 7th Ed. Addison Wesley, Indian Edition, 2000.
4. Elmasri & Navathe : Fundamentals of Database Systems/Oracle 9i Programming 5th Ed. Pearson, 2009.

Course Outcomes

1. To understand fundamental concepts, principles and applications of database system.
2. To demonstrate database related programming languages and perform the basics of commercial relational systems.
3. To apply the concepts of normalization on database design.
4. To Design and Implement a small database project, considering the issues like concurrency control, recovery and security.
5. Comprehend contemporary issues related to industry and government related to database domain.

Course Code	:	**CSPC27**
Course Title	:	**Software Engineering**
Number of Credits	:	**4**
Prerequisites	:	
Course Type	:	**PC**

Course Learning Objectives:

1. Study the current software engineering techniques and examines the software life-cycle, including software specification, design implementation, testing and maintenance.
2. Present software engineering methodologies for the development of Quality, cost-effective, schedule-meeting software.
3. Develop an understanding of ethical and professional issues related to Software Project Delivery.

Course Content:

1. **Introduction**: Introduction of Software (SW), Type of Software, SW Components: Process, People, Project, Product, Software crisis, Software Process Models: Details of People involve in each Process, SDLC methods/models: Build & Fix, Waterfall, Prototype (Evolutionary & Throw-away), Iterative, Incremental iterative, Spiral, RAD, Agile methodology.

2. **Requirement Analysis & Specifications**: Requirement Analysis, Requirement Specification, Approaches to Requirement analysis, Specifying Behavioural & Non-Behavioural Requirements, SRS Components & various User's of SRS. Introduction of Requirement Specification: Dataflow(DF) Diagram, Data dictionaries, Entity-Relationship (ER) diagram, Object Diagram etc., Requirement Validation.

3. **Software Design and Testing:** Design Architecture and Patterns, Modularity, Function oriented design, Object Oriented Design, Software Testing: Software Testing Strategy and Techniques, Functional testing, Structural testing, Debugging and testing tools, SW/HW reliability, Reliability concepts and models, Reliability allocation, Software Maintenance: Introduction to SW Maintenance and types, SW Maintenance models: Re-engineering & Forward Engineering.

4. **Software Project Planning**: Role of Software Project Planning, Estimation method, Estimation of Effort & Schedule, Software Metrics: Introduction to Size metrics, Data structure metrics, information flow metrics, entropy-based measures, metric analysis. Basic COCOMO, Intermediate COCOMO, Detailed COCOMO, Quality Planning, Planning Parameter, Quality Defect Removal Cycle, Role of Risk Analysis.

Reference Books

1. R.S. Pressman, Software Engineering – A Practitioner's Approach, 6th Edition, TMH, 2013.
2. Ian Sommerville, Software Engineering, 8th Edition, Addison Wesley, 2009.
3. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing, 2010.
4. K.K.Aggarwal, Yogesh Singh: Software Engineering, New Age International Ltd, 2001.

Course Outcomes

1. Able to apply the concepts and choose an appropriate SDLC process model for user requirements.
2. To analyze requirement techniques like Data flow diagram, Entity relationship diagram etc.

3. Understanding the concept of Software Design and emphasizing upon various software metrics used for analyzing the software.
4. Demonstrate various testing methodologies and debugging tools for a prototype software.
5. Design various software reliability measures to assess the quality of software in case of various faults and failures.

Course Code	**:**	**CSPC29**
Course Title	**:**	**Computer Organization and Architecture**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSPC10**
Course Type	**:**	**PC**

Course Learning Objectives:

1. Working of Computer Systems & its basic principles
2. Concepts of processor and control design.
3. Concepts of pipelining techniques.
4. Memory hierarchy and its organization.
5. Concept of I/O devices and working.

Course Content

1. Introduction

Introduction to digital electronics: combinational circuits and sequential circuits. Basic Machine Principle, Structure and representation of real world data. Subroutine, Branching & Macro facility.

2. Processor Design

Processor Organization, Information representation and Number format, Instruction cycle and Instruction format, Addressing modes, Arithmetic operation, timed point addition, subtraction, multiplication and division, ALU design, Parallel processing – Performance consideration, Pipeline processor.

3. Control Design

Instruction sequencing and Interpretation, Hardware Control design method and Microprogrammed Control.

4. Memory organization

Memory device characteristic, Random access and serial access memories, Virtual memory – memory hierarchies, Page replacement policies, Segments, pages and file organization, High speed memories – cache and associative memory.

5. System Organization

Programmed I/O, DMA and interrupts, I/O processors & CPU – I/O interaction.

Reference Books:

1. M.M. Mano: Computer System Architecture, 3rd Ed. PHI.
2. J.P. Hayes: Computer Architecture and Organization, 3rd Ed. TMH, 1999.
3. C.W. Gear: Computer organization and Programming, TMH.
4. T.C. Bartee: Digital Computer Fundamental, TMH.
5. A. S. Tanenbaum: Computer System Organization, PHI.

Course outcomes:

1. Understand the Computer System concepts.
2. Understand the organization of a computer system in terms of its main components
3. Understand the processor and control design of a computer system.
4. Understand the various types of memory.

5. Understand input/output mechanisms.

Course Code	**:**	**CSPC20**
Course Title	**:**	**Operating Systems**
Number of Credits	**:**	**5**
Prerequisites	**:**	
Course Type	**:**	**PC**

Course Learning Objectives:

1. To understand the services and design of an operating system.
2. To understand the structure and organization of file system
3. To understand the process states and various concepts such as scheduling and synchronization related with it.
4. To understand different memory management approaches.
5. Students should be able to use system calls for managing processes, memory and file system.
6. Students should understand the data structures and algorithms for implementation of OS.

Course Content:

1. Computer system architecture and organization, Introduction and evolution of OS, Introduction to distributed OS, Real time systems and multimedia systems. OS structures: OS services, system calls and programs, OS design and implementation. Processes: Process concept, scheduling policies, algorithms, multilevel queuing, operations on process, Inter-process communication. Threads: multithreading models and threading issues. CPU scheduling: Criteria and algorithms, multiprocessor and thread scheduling.

2. Process synchronization: critical sections, classical two process and n-process solutions, hardware primitives for synchronization, semaphores, monitors, classical problems in synchronization (producer-consumer, readers-writer, dining philosophers, etc.). Deadlocks: modeling, resource allocation, characterization, prevention and avoidance, detection and recovery.

3. Memory management: Swapping, contiguous memory allocation, paging, multilevel paging, segmentation, demand paging, page replacement algorithms, allocation of frames, thrashing, working set model. Input/Output: I/O system and services, device controllers and device drivers, disks, scheduling algorithms and management.

4. File system interface: access methods, access control, directory structures, file organization, file sharing and protection. system performance, protection and security, access matrix, Security breaches, types of attacks, attack prevention methods, security policy and access control, OS design considerations for security, access, policy and access control, OS design considerations for security, access control lists and OS support, internet and general network security.

Reference Books:

1. A. Silberschatz, Peter B. Galvin and G. Gagne, "Operating System Concepts," (6th or newer edition), Wiley.
2. H. Brinch, "Operating System Principles," Prentice Hall of India.
3. Dhamdhere, "Systems programming & Operating systems," TataMcGraw Hill.
4. A. N. Habermann, "Introduction to Operating System Design," Galgotia publication, New Delhi.
5. A.S. Tanenbaum, "Modern Operating Systems," Prentice Hall of India.

Course outcomes:

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

1. Understand functions, structures and history of operating systems.
2. Able to know the design issues associated with operating systems.

3. Master various process management concepts such as scheduling, synchronization, multithreading and deadlocks.
4. Understand the various concepts associated with memory management such as virtual memory, demand paging, page replacements algorithms.
5. Be familiar with various protection and security mechanisms.

Course Code	**:**	**CSPC22**
Course Title	**:**	**Microprocessor and Interfacing**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSPC10, CSPC29**
Course Type	**:**	**PC**

Course Learning Objectives:

1. To provide an introduction to the basic principles and applications of microprocessor to solve real life problems.
2. To give an expose of assembly language programming.
3. To make familiar with the implementation of various interfaces with microprocessor.

Course Content:

1. **Introduction & Architecture**
   History of microprocessors, 8086 architecture, internal operation, Machine language instructions, Instruction execution time (Timing Diagram: Fetch Cycle, Execution Cycle).

2. **Assembly Language Programming**
   Instruction format, Instruction Types (data transfer instruction, arithmetic instructions, branch instruction, NOP & HLT instructions, flag manipulation instruction, logical instruction, shift and rotate instruction, String instructions), Assembler directions and operators.

3. **Memory Interfacing and Modular Programming**
   Memory Interfacing (Static RAMs & ROMs), Stacks, Procedures, macros – local labels and nested macros.

4. **I/O Interface and I/O Programming**
   I/O consideration, programmed I/O block transfer, Serial communication, asynchronous, synchronous, physical, 8251A; Parallel communication: 8255 A, 16-bit bus interface, Basic Interrupt processing, Interrupt and interrupt routines, 8259A Programmable Interrupt Controller, DMA with controller (8237), 8254 Programmable interval Timer (8253/8254), Interfacing to A/D and D/A converters, Stepper motor interfacing.

Reference Books:

1. Liu and Gibson, Microcomputer Systems: 8086/8088 family: Architecture, Programming and Design, PHI.
2. D.V. Hall, Microprocessors and Interfacing, TMH.
3. A. K. Ray & K.M. Bhurchandi, Advanced Microprocessors and Peripherals: Architecture, Programming and interfacing, TMH.
4. Bray, the Intel Microprocessor 8086/8088-Pentium: Architecture, Programming and interfacing, PHI
5. James L. Antonakos, The Intel Microprocessor family H/W and S/W Principles and Applications, Cengage Learning

Course outcomes:

1. Develop an understanding of basic concepts of microprocessors.
2. Understand the instruction set of 8086.
3. Able to apply the knowledge of assembly language to solve various problems.
4. Grasp an understanding of various peripheral device interfaces with 8086.
5. Able to design and implement various interfaces in real life different applications.

Course Code	**:**	**CSPC24**
Course Title	**:**	**Unix and Linux Programming**
Number of Credits	**:**	**3**
Prerequisites	**:**	
Course Type	**:**	**PC**

Course Learning Objective:

1. To introduce the Unix/Linux operating system and shell programming basics.
2. Knowledge of Unix/Linux tools for file handling.
3. Analysis of C/C++ environment of Unix/Linux and learn basic of Pythan programming
4. Introduction of process manipulation and learning password management for user privileged in Linux.

Course Content:

1. **Linux Startup and Shell Programming**
   User accounts, accessing linux- starting and shutting process, Logging in and Logging out, Command line, simple command, Unix file system: Linux/Unix files, inodes and structure and file system related commands, Shell as command processor, shell variables, creating command substitution, scripts, functions, conditionals, loops, customizing

2. **Regular Expressions and Filters**
   Introducing regular expressions patterns, syntax, character classes, quantifiers, introduction to egrep, sed, programming with awk and perl

3. **The C/C++ Environment**
   The C compiler, vi editor, compiler options, managing projects, memory management, use of makefiles, dependency calculations, memory management- dynamic and static memory, building and using static and dynamic libraries, using ldd, soname, dynamic loader, debugging with gdb, introduction to Pythan programming

4. **Processes in Linux**
   Processes, starting and stopping processes, initialization processes, rc and init files, job control- at, batch, cron, time, network files, security, privileges, authentication, password administration, archiving, Signals and signal handlers, Linux I/O system.

Reference Books:

1. Sumitabha Das: Your Unix- The Ultimate Guide, TMH, 2012
2. Behrouz A. Forouzan, Richard F. Gilberg: UNIX and Shell Programming, Brooks/Cole-Thomson Learning (CENGAGE Learning, India) , 2009
3. John Goerzen: Linux Programming Bible, IDG Books, New Delhi, 2000
4. Neil Matthew, Richard Stones: Beginning Linux Programming, Wrox-Shroff, 2007
5. Welsh & Kaufmann: Running Linux, O'Reiley & Associates, 2000
6. Guido van Rossum: An Introduction to Python, Network Theory Ltd, 2011
7. The Python Tutorial Online Book (http://docs.python.org/3/tutorial/index.html)

Course Outcomes (CO's)

1. Able to understand the Unix/Linux operating system, installation procedures, basic utilities command, and basics of shell programming.
2. Ability to handle the files or records through various Unix/Linux tools by using regular expression.
3. Able to learn C/C++ environment, Makefile, libraries, linking, debugging methodologies and apply the advantages of Pythan programming.
4. Able to manipulate the processes and manage the networks securely in Linux.

Course Code	**:**	**CSPC26**
Course Title	**:**	**Computer Networks**
Number of Credits	**:**	**5**
Prerequisites	**:**	
Course Type	**:**	**PC**

Course Learning Objectives:

1. Understand computer network basic, different models used for study of computer networks, ability to identify different designs, understanding of the issues surrounding wired and wireless Networks.
2. Design, calculate, and apply subnet masks to fulfill networking requirements and building the skills of routing mechanisms.
3. Analyse the features and operations of various application layer protocols such as Http, DNS, SMTP and FTP.
4. Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies
5. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Course content

1. Introduction

Network Functions, Network Topology, Network Services, Switching Approaches, Transmission media and systems, multiplexing and signaling techniques, Error detection and correction, Internet checksum algorithm.

2. Layered Architectures

Examples, OSI Reference Model, Overview of TCP/IP architecture, Socket system calls, SNMP, Electronic Mail.

3. Peer-to-Peer Protocols

Protocols, Service Models and End-to-End requirements, ARQ, Sliding Window Protocols, RTP, HDLC, PPP protocols, Statistical Multiplexing.

4. MAC and LAN Protocols

Multiple access communication, Random Access-ALOHA, Slotted-ALOHA, CSMA, CSMA-CD, LAN Standards – Ethernet, Fast Ethernet & Gigabit Ethernet, Bluetooth and WiMax standards.

5. Packet Switching Networks

Packet network topology, Datagrams and Virtual Circuits – Structure of Switch / Router, Connectionless and Virtual Circuit packet Switching, Traffic management and QoS – FIFO, Priority Queues, Fair Queuing, MPLS.

6. TCP/IP

Architecture, Internet protocols – IP packet, Addressing, Subnet addressing, IP routing, CIDR, ARP, RARP, ICMP, Reassembly, IPv6, UDP, Transmission Control Protocol – TCP, DHCP, Mobile IPv6, Internet Routing protocols, Multicast Routing.

BOOKS

1. Leon Garcia and IndraWidjaja: Communication Networks – Fundamental Concepts and Key Architectures, TMH, 2000.
2. A.S. Tanenbaum: Computer Networks, Fourth Edition, Pearson Education, 2003.
3. Forouzan: Data Communications and Networks, Fourth Edition, McGraw Hill, 2007.

4. William Stallings: Data and Computer Communications 5/e, PHI.

Course Outcomes:

1. Understand computer network basic, different models used for study of computer networks, ability to identify different designs, understanding of the issues surrounding wired and wireless Networks.
2. Design, calculate, and apply subnet masks to fulfil networking requirements and building the skills of routing mechanisms.
3. Analyse the features and operations of various application layer protocols such as Http, DNS, SMTP and FTP.
4. Analyse the requirements for a given organizational structure and select the most appropriate networking architecture and technologies
5. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Programme Electives – I and II

Course Code	**:**	**CSPE20**
Course Title	**:**	**Advanced Java Programming**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSIR11, CSPC23**
Course Type	**:**	**PE**

Course Learning Objectives:

1. How to interact with a relational database using java programming language.
2. How to write generic java programs.
3. Writing server side programs using java technology.

Course Content:

1. JDBC Architecture, a Relational Database Overview, Processing SQL Statements with JDBC Establishing a Connection, Connecting with DataSource Objects, Handling SQLExceptions, Retrieving and Modifying Values from Result Sets, Using Prepared Statements, Using Transactions, Using RowSet Objects

2. Generics & Collection Framework APIs. Introduction to Design Patterns: the Factory Design Pattern, the Singleton Design Pattern..

3. Why use Servlets & JSPs: an introduction to web servers & clients, HTML, HTTP Protocol, HTTP GET and POST requests, HTTP responses. Web App Architecture: high-level overview. A Model-View-Controller (MVC) overview and example, life cycle of a servlet, request & response objects, Init Parameters and ServletConfig, JSP init parameters, Context init parameters, attributes and listeners, session management.

4. Create a simple JSP using "out" and a page directive, JSP expressions, variables, and declarations, implicit objects, The Lifecycle and initialization of a JSP, other directives. Scriptless JSP: Standard actions, Expression Language, The EL implicit objects & EL functions, using JSTL.

Reference Books:

1. Dietel & Deitel, *Java How to Program*, Pearson Education, 10th Ed., 2015.
2. Bryan Basham, Kathy Sierra, Bert Bates, *Head First Servlets & JSPs* , O'REILLY, 2nd Ed., 2008.
3. Eric Freeman , Elisabeth Freeman, Kathy Sierra and Bert Bates, *Head First Design Patterns*, O'REILLY, 1st Ed., 2004.

Course outcomes:

1. Able to interact with a database using java-programming language.
2. Able to write generic programs with the usage of collection framework APIs
3. Able to design a web application using various J2EE constructs such as servlets and JSPs.

Course Code	**:**	**CSPE22**
Course Title	**:**	**Programming using Python**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSPC12, CSPC23**
Course Type	**:**	**PE**

Course Learning Objectives:

1. Building robust applications using Python programming language's features.
2. Understanding the usage of Python libraries.
3. Building multithreaded, platform-independent and GUI based python applications for business problems.

Course Content:

1. The concept of data types; variables, assignments; immutable variables; numerical types; arithmetic operators and expressions; comments in the program; understanding error messages; Conditions, boolean logic, logical operators; ranges; Control statements: if-else, loops (for, while); short-circuit (lazy) evaluation; Strings and text files; manipulating files and directories, os and sys modules; text files: reading/writing text and numbers from/to a file; creating and reading a formatted file (csv or tab-separated); String manipulations: subscript operator, indexing, slicing a string.

2. Lists, tuples, and dictionaries; basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and replacing values; traversing dictionaries; Design with functions: hiding redundancy, complexity; arguments and return values; formal vs actual arguments, named arguments.

3. Simple Graphics and Image Processing: "turtle" module; simple 2d drawing - colors, shapes; digital images, image file formats, image processing: Simple image manipulations with 'image' module (convert to bw, greyscale, blur, etc). Classes and OOP: classes, objects, attributes and methods; defining classes; design with classes, data modeling; persistent storage of objects; inheritance, polymorphism, operator overloading (_eq_, _str_, etc); abstract classes; exception handling, try block

4. Graphical user interfaces; event-driven programming paradigm; tkinter module, creating simple GUI; buttons, labels, entry fields, dialogs; widget attributes - sizes, fonts, colors layouts, nested frames.

Reference Books:

1. T.R. Padmanabhan, Programming with Python, Springer, 1st Ed., 2016.
2. Kenneth Lambert, Fundamentals of Python: First Programs, Cengage Learning,, 1st Ed., 2012.

Course outcomes:

1. Write python programs that solve simple business problems.
2. Create python applications that are robust and multithreaded.
3. Write simple GUI interfaces for a program to interact with users, and to understand the event-based GUI handling principles in python.

Course Code	**:**	**CSPE24**
Course Title	**:**	**Number Theory and Cryptography**
Number of Credits	**:**	**4**
Prerequisites	**:**	
Course Type	**:**	**PE**

Course Learning Objectives:

1. To understand Mathematical foundations of Cryptography theory.
2. To analyze the robustness of Cryptosystems.
3. To design robust cryptosystem for real time applications.
4. To develop prototype Cryptosystems and demonstrate their performance.

Course content

1. **Elementary Number Theory:** Divisibility, Division Algorithm, Euclidean Algorithm; Congruences, Complete Residue systems, Reduced Residue systems; Fermat's little theorem, Euler's Generalization, Wilson's Theorem; Chinese Remainder Theorem, Generalized Chinese Remainder Theorem-Euler Phi-function, multiplicative property; Finite Fields, Primitive Roots; Quadratic Residues, Legendre Symbol, Jacobi Symbol; Gauss's lemma, Quadratic Reciprocity Law.

2. **Primality Testing and Factorization:** Primality Tests; Pseudoprimes, Carmichael Numbers; Fermat’s pseudoprimes, Euler pseudoprimes; Factorization by Pollard’s Rho method; Simple Continued Fraction, simple infinite continued fractions; Approximation to irrational numbers using continued fractions; Continued Fraction method for factorization.

3. **Private Key Cryptosystems:** Classical methods: Caesar cipher, Vigenere cipher, one-time pad and others; Block ciphers, Fiestel structure, DES, AES and its security properties; Linear Cryptanalysis, Differential Cryptanalysis; Hash functions: MD5, SHA-1 and its properties

4. **Public Key Cryptosystems:** Traditional Cryptosystem, limitations; Public Key Cryptography; Diffie-Hellmann key exchange; Discrete Logarithm problem; One-way functions, Trapdoor functions; RSA cryptosystem; Digital signature schemes; Digital signature standards; RSA signature schemes; Knapsack problem; ElGamal Public Key Cryptosystem; Attacks on RSA cryptosystem: Common modulus attack; Homomorphism attack, timing attack; Forging of digital signatures; Elliptic curve cryptography: Cubic Curves, Singular points, Discriminant; Introduction to Elliptic Curves, Geometry of elliptic curves over reals; point at infinity; Addition of two points; Elliptic Curves over finite fields, ElGamal Public Key Cryptosystem for elliptic curves.

Text Books, and/or reference material

1. A Course in Number Theory and Cryptography, Neal Koblitz, (Springer 2006)
2. An Introduction to Mathematical Cryptography, Jill Pipher, Jeffrey Hoffstein, Joseph H. Silverman (Springer, 2008)
3. An Introduction to theory of numbers, Niven, Zuckerman and Montgomery, (Wiley 2006)
4. Elliptic curves: Number theory and cryptography, Lawrence C. Washington, (Chapman & Hall/CRC 2003)
5. An Introduction to Cryptography, R.A. Mollin (Chapman & Hall, 2001)
6. Rational Points on Elliptic Curves, Silverman and Tate (Springer 2005)
7. Guide to elliptic curve cryptography Hankerson, Menezes, Vanstone (Springer, 2004)
8. Elementary Number Theory, Jones and Jones (Springer, 1998).

Course outcomes

1. To understand Mathematical foundations of Cryptography theory.
2. To analyze the robustness of Cryptosystems.
3. To design robust cryptosystem for real time applications.
4. To develop prototype Cryptosystems and demonstrate their performance.

Course Code	**:**	**CSPE26**
Course Title	**:**	**Efficient Coding for Problem Solving and Competitive Programming**

Number of Credits	**:**	**4**
Prerequisites	**:**	**CSPC12, CSPC21** **Good practice of programming and complexity analysis in Data Structures & Algorithms**
Course Type	**:**	**EPR**

Course Learning Objectives:

1. Identifying multiple alternate coding solutions of problems discussed in programming tests and Competitive Problems (CP) and developing understanding of their efficiency issues.
2. Designing and developing brute-force solutions followed by time/space efficient solutions for CP problems.
3. Learning commonly used problem solving approaches and applying them for coding the solutions for new problems.

Course Content:

1. Overview of Efficient Coding for Problem Solving and CP: Introduction to competitive programming (CP); revisit of complexity analysis; introduction to online platforms such as codechef, codeforces etc and online submission; constraints during CP, online testing process and common errors such as TLE; use of STL
2. Use of Number Theory for problem-solving: reducing time/space complexity of brute force coding solution of Sieve Method, Inverse Module, Euclidian Method of factorization; efficient coding for Permutation Combination; XORing based and pattern based solutions,
3. Coding for Optimizing time and Space using Sequential Storage: two pointer approach; problem-solving using arrays and strings such as rotation on sorted arrays, duplicate removal, string matching algorithms; Kadane’s algo, stacks, priority-queues and hashing based efficient coding; median based problems and alternate solutions.
4. Applying Non-Linear Data Structures for real-life problems: design of efficient solutions for problems such as finding loops in a linked list, memory efficient DLL, block reversal in LL; problem solving using trees and binary trees, Catalan numbers, applications of graphs, spanning tree and path algos for CP problems with reduced time/space complexity
5. CP Problem Solving using Advanced Topics: concept of disjoint sets and their efficient representation, algorithmic approaches such as Greedy, Backtracking, Dynamic Programming and applying them for CP problems using bottom-up dynamic programming.

Text Books:

1. Karumanchi: Data Structures and Algorithms Made Easy, CareerMonk Publications, 2017
2. G L McDowell: Cracking the Coding Interviews, CareerCup Publications, 2016
3. geekforgeeks.org
4. codechef.com, spoj.com, codeforces.com, interviewbit.com

Course Outcomes:

1. Understanding the issues of online platforms and Competitive Programming (CP) and developing brute force coding for commonly asked CP problems
2. Analyzing the space and time complexity of brute force solutions and designing efficient solutions
3. Evaluating the applicability of suitable algorithmic approaches to solve relevant CP problems

4. Creating efficient solutions of CP problems using the learnt algorithmic approaches

Course Code	**:**	**CSPE60**
Course Title	**:**	**Stochastic Processes and Queuing Theory**
Number of Credits	**:**	**4**
Prerequisites	**:**	**MAIR13**
Course Type	**:**	**PE**

Course Learning Objectives:

1. It provides an essential base for mathematical modelling which is normally used to solve the problems of pattern recognition and machine learning.
2. It is used in the research of various science and engineering problem.

Course Content:

1. Random variables, Functions of random variables, joint1y distributed random variables. Expectation, Expectation of functions of more than one random variable, Covariance and Correlation, Auto and Cross Correlation, Correlation Co-efficient, Parameter Estimation – Maximum Likelihood Estimation and Maximum a Posteriori.

2. Stochastic Processes, Classification of stochastic processes, Stationary Random Processes- First order, second order and nth order, Strict sense stationary process, Ergodic Process, the Bernoulli process, The Poisson process, Renewal process, Advanced Renewal Theory, Renewal Function, Alternating Renewal Processes .

3. Markov Chains, Computation of n-step transition probabilities, Chapman-Kolmogorov theorem, Classification of states of a Markov Chain, Distribution of times between state changes, Irreducible finite chains with aperiodic states, Birth and Death processes, Pure Birth Process, Pure Death Process, Non-Birth-Death Processes, Analysis of program execution time..

4. Introduction to Queuing Theory, General Queuing, Components of a queuing system, Deterministic Queues, (M/M/1):(∞/FIFO)-Single server with infinite capacity, Extension to (M/M/2): (∞/FIFO), (M/M/k) : (∞/FIFO)- Multiple server with infinite capacity, Engset Loss Formula, M/G/1 queuing system, M/G/1 with non-FIFO disciplines, M/G/∞, Queues with time-varying arrival rate.

Reference Books:

1. A. Populis and S.U. Pillai, “Probability, Random Variables, and Stochastic Process”, 4th Edition, McGraw-Hill.
2. S. Palaniammal, “Probability and Queueing theory.” Prentice Hall of India.
3. V. Sundarapandian**, “**Probability, Statistics and Queueing Theory.” Prentice Hall of India.

Course outcomes

1. Develop an understanding to the basic concepts of Random Processes, Expectations and Parameter Estimation.
2. Able to understand and apply the Stochastic processes to Science and Engineering problems and applications.
3. Able to calculate the n-step transition probabilities for any Markov chain and understand about the birth and death of processes.
4. Able to apply Markov chain & Birth Death process to real life problems.
5. Develop an understanding of various Queuing Systems.

Course Code	**:**	**CSPE62**
Course Title	**:**	**Software Project Management**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSPC27**

Course Type	**:**	**PE**

Course Learning Objectives:

Understand the different activities in software project development i.e, planning, design and management.

Course content:

1. **Conventional Software Management**.
   Evolution of software economics. Improving software economics: reducing product size, software processes, team effectiveness, automation through software environments. Principles of modern software management.

2. **Software Management Process**
   Framework,: Life cycle phases- inception, elaboration, construction and training phase. Artifacts of the process- the artifact sets, management artifacts, engineering artifacts, pragmatics artifacts. Model based software architectures. Workflows of the process. Checkpoints of the process.

3. **Software Management Disciplines**
   Iterative process planning. Project organisations and responsibilities. Process automation. Project control and process instrumentation- core metrics, management indicators, life cycle expections. Process discriminants.

Books

1. Software Project management, Walker Royce, Addison Wesley, 1998.
2. Project management 2/e, Maylor.
3. Managing the Software Process, Humphrey.
4. Managing global software Projects, Ramesh, TMH, 2001.

Course Outcomes:

1. Understanding the evolution and improvement of software economics according to the basic parameters and transition to the modern software management.
2. Learning the objectives, activities and evaluation criteria of the various phases of the life cycle of software management process.
3. Gaining knowledge about the various artifacts, workflows and checkpoints of the software management process and exploring the design concept-using model based architecture from technical and management perspective.
4. Develop an understanding of project planning, organization, responsibilities, automation and control of the processes to achieve the desirable results.

Course Code	**:**	**CSPE64**
Course Title	**:**	**Data Mining and Data Warehousing**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSPC25**
Course Type	**:**	**PE**

Course Learning Objectives:

1. Develop an understanding on the basic concepts, principles and applications of data mining and warehousing, also able to identify the scope and necessity of system.
2. Describe the theoretical constructs and core processes of data mining and warehousing, also demonstrate the importance of paradigms from the fields of Artificial Intelligence and Machine Learning to data mining.
3. To understand the strengths and limitations of various data mining and warehousing models and techniques in solving the real world problems.
4. Able to design and analyse, a lightweight prototype or simulation that supports the concept of data mining and warehousing.

5. Comprehend contemporary issues related to research domain, industry and government, also able to inculcate professional, ethical issues and social responsibilities.

Course content

1. Introduction : Fundamentals of data mining, Data Mining Functionalities, Classification of Data Mining systems, Major issues in Data Mining, Data Warehouse and OLAP Technology for Data Mining Data Warehouse, Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Further Development of Data Cube Technology, From Data Warehousing to Data Mining, Data Preprocessing : Needs Preprocessing the Data, Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization and Concept Hierarchy Generation.

2. Data Mining Primitives, Languages, and System Architectures : Data Mining Primitives, Data Mining Query Languages, Designing Graphical User Interfaces Based on a Data Mining Query Language Architectures of Data Mining Systems, Concepts Description : Characterization and Comparison : Data Generalization and Summarization-Based Characterization, Analytical Characterization: Analysis of Attribute Relevance, Mining Class Comparisons: Discriminating between Different Classes, Mining Descriptive Statistical Measures in Large Databases.

3. Mining Association Rules in Large Databases : Association Rule Mining, Mining Single-Dimensional Boolean Association Rules from Transactional Databases, Mining Multilevel Association Rules from Transaction Databases, Mining Multidimensional Association Rules from Relational Databases and Data Warehouses, From Association Mining to Correlation Analysis, Constraint-Based Association Mining. Classification and Prediction: Issues Regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification, Classification by Backpropagation, Classification Based on Concepts from Association Rule Mining, Other Classification Methods, Prediction, Classifier Accuracy.

4. Cluster Analysis Introduction : Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning Methods, Density-Based Methods, Grid-Based Methods, Model-Based Clustering Methods, Outlier Analysis. Mining Complex Types of Data: Multimensional Analysis and Descriptive Mining of Complex, Data Objects, Mining Spatial Databases, Mining Multimedia Databases, Mining Time-Series and Sequence Data, Mining Text Databases, Mining the World Wide Web.

Text Book:

1. Data Mining – Concepts and Techniques - Jiawei Han & Micheline Kamber Harcourt India.

References:

1. Data Mining Introductory And Advanced Topics –Margaret H Dunham, Pearson Education
2. Data Mining Techniques – Arun K Pujari, University Press.
3. Data Warehousing In the Real World – Sam Anahory & Dennis Murray. Pearson Edn Asia.
4. Data Warehousing Fundamentals – Paulraj Ponnaiah Wiley Student Edition.
5. The Data Warehouse Life Cycle Tool Kit – Ralph Kimball Wiley Student Edition.

Course Outcomes:

1. Develop an understanding on the basic concepts, principles and applications of data mining and warehousing, also able to identify the scope and necessity of system.
2. Describe the theoretical constructs and core processes of data mining and warehousing, also demonstrate the importance of paradigms from the fields of Artificial Intelligence and Machine Learning to data mining.
3. To understand the strengths and limitations of various data mining and warehousing models and techniques in solving the real world problems.
4. Able to design and analyse, a lightweight prototype or simulation that supports the concept of data mining and warehousing.
5. Comprehend contemporary issues related to research domain, industry and government, also able to inculcate professional, ethical issues and social responsibilities.

Course Code	**:**	**CSPC31**
Course Title	**:**	**Advanced Data Structures and Algorithms**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSPC11**
Course Type	**:**	**PC**

Course Learning Objectives:

1. To develop the understanding of advanced data structures.
2. To develop the understanding of algorithmic design paradigms.

Course Content

1. **Advanced Data Structures**
   Threaded binary trees, Binomial heaps, Fibonacci heaps, data structures for disjoint sets, Balanced trees- AVL, red-black trees

2. **String Matching**
   String-matching algorithm, Rabin-Karp algorithm, String matching with automata, Knuth-Morris-Pratt algorithm.

3. **NP – Hard & NP – Complete problems**
   Concepts of NP-hard & NP complete problems, Clique Decision problem, Node Cover decision problem, Travelling Salesperson decision problem.

4. **Advanced Algorithms**
   Introduction to approximation algorithms, absolute approximation, ε-approximation, polynomial time approximations, randomized algorithms.

Suggested Books:

1. E Horowitz and S. Sahni, Fundamentals of Data Structures, Universities Press.
2. Cormen, Leiserson and Rivest, Introduction to Algorithms, 3/e, PHI.
3. Horowitz, Ellis and Sahni, Fundamentals of Computer Algorithms, Universities Press..

Reference Books:

1. Oded Goldreich. P, NP, and NP-Completeness: The Basics of Computational Complexity, Cambridge University Press, 2010.
2. Anne Benoit, Yves Robert, Frédéric Vivien. A Guide to Algorithm Design**:** Paradigms, Methods, and Complexity Analysis, Taylor & Francis, 2013.
3. A.V. Aho, J.E. Hopcroft, and J.D. Ullman, Data Structures and Algorithms, Addison Wesley, Reading Massachusetts, USA, 1983.
4. Donald Knuth. The Art of Computer Programming: Fundamental Algorithms, Third Edition. Addison-Wesley, 1997.
5. Donald Knuth. The Art of Computer Programming Volume 3: Sorting and Searching, Third Edition. Addison-Wesley, 1997. ISBN 0-201-89685-0.

Course Outcome:

1. Compare, contrast, and apply the key algorithmic design paradigms: brute force, divide and conquer, decrease and conquer, transform and conquer, greedy, dynamic.
2. Compare, contrast, and apply key data structures: trees, lists, stacks, queues, hash tables, and graph representations.
3. Compare, contrast, and apply algorithmic tradeoffs: time vs. space, deterministic vs. randomized, and exact vs. approximate.
4. Implement, empirically compare, and apply fundamental algorithms and data structures to real-world problems.

Course Code	**:**	**CSPC33**
Course Title	**:**	**Soft Computing**
Number of Credits	**:**	**4**
Prerequisites	**:**	
Course Type	**:**	**PC**

Course Learning Objectives:

1. Motivation and historical background of Soft Computing.
2. Application of Fuzzy logic.
3. Biologically inspired algorithm such as neural networks, genetic algorithms, ant colony optimization, and bee colony optimization.
4. Hybrid systems of neural network, genetic algorithms and fuzzy systems.

Course content:

1. **Soft Computing and Artificial Intelligence:** Introduction of Soft Computing, Soft Computing vs. Hard Computing, Various Types of Soft Computing Techniques, Applications of Soft Computing, AI Search Algorithm, Predicate Calculus, Rules of Interference, Semantic Networks, Frames, Objects, Hybrid Models.

2. **Artificial Neural Networks and Paradigms:** Introduction to Neuron Model, Neural Network Architecture, Learning Rules, Perceptrons, Single Layer Perceptrons, Multilayer Perceptrons, Back propagation Networks, Kohnen'sself organizing networks, Hopfield network, Applications of NN.

3. **Fuzzy Logic:** Introduction, Fuzzy sets and Fuzzy reasoning, Basic functions on fuzzy sets, relations, rule based models and linguistic variables, fuzzy controls, Fuzzy decision-making, applications of fuzzy logic.

4. **Genetic Algorithms and Swarm Optimizations:** Introduction, Genetic Algorithm, Fitness Computations, Cross Over, Mutation, Evolutionary Programming, Classifier Systems, Genetic Programming Parse Trees, Variants of GA, Applications, Ant Colony Optimization, Particle Swarm Optimization, Artificial Bee Colony Optimization.

Suggested Books :-

1. Simon S. Haykin, Neural Networks, Prentice Hall, 2nd edition.
2. B. Yegnanrayana , “Artificial Neural Networks” , PHI.
3. Jacek M. Zurada, Introduction to Artificial Neural Systems, Jaico Publishing House,1994
4. Zimmermann, “Fuzzy Set Theory and its Application”, 3rd Edition.
5. Jang J.S.R., Sun C.T. and Mizutani E, "Neuro-Fuzzy and Soft computing", Prentice Hall, 1998.
6. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", McGraw Hill, 1997.
7. D.E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley, N.Y, 1989.

Course Outcomes

1. Understand various biologically inspired algorithms.
2. Apply different intelligent algorithms in classification, prediction, optimization, pattern recognition applications.
3. Design hybrid system incorporating neural network, genetic algorithms, fuzzy systems.

Course Code	**:**	**CSPC35**
Course Title	**:**	**Software Development using UML and Agile Methodology**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSPC23**
Course Type	**:**	**PC**

Course Learning Objectives:

The objective of this course is to prepare students for software development using UML and agile methodology for object oriented analysis and design

Course content

1. **UML & Object-Oriented Modeling [1] [3]**

   An Overview of UML, architecture, UML Modeling Concepts – system models & views, datatypes, classes, objects, event & messages, object-oriented modeling; Modeling Relations- association, aggregation composition, generalization, dependency and structural diagrams, Requirement Elicitation- introduction, concepts & activities[3].

   Object oriented analysis design using diagrams such as use-case diagrams, class diagrams, sequence diagrams, collaboration diagrams, activity diagrams, object diagrams, interaction diagrams and statechart diagrams.

2. **Fundamentals of Agile [4]**

   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

3. **Agile Scrum Framework [2]**

   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

4. **Agile Software Design and Development [2] [4]**

   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

Reference Books:

1. Bernd Bruegge, Allen H. Dutoit, Object-Oriented Software Engineering Using UML, Patterns and Java, Pearson, 3rd Edition, 2013.
2. Ken Schawber, Mike Beedle, Agile Software Development with Scrum, Pearson, 2001.
3. Grady Booch, James Rumbaugh, Ivar Jacobson, Unified Modeling Language User Guide, Addison Wesley, 2nd Edition, 2012.
4. Robert C. Martin, Agile Software Development- Principles, Patterns and Practices, Prentice Hall, 2013.

Course Outcomes:

On completion of the course the student will be able to:

1. Understand the concepts of Unified Modeling Language (UML) and UML diagrams during object oriented analysis and design.
2. Software development and process control using agile and scrum methodology.

Course Code	:	**CSPC37**
Course Title	:	**Automata Theory**
Number of Credits	:	**4**
Prerequisites	:	**MAIR13**
Course Type	:	**PC**

Course Learning Objectives:

The objective of this course is to give students a clear understanding of concepts formal languages, grammars and formal methods of computation.

Course Content:

1. Introduction to Finite State Machine, Moore and Mealy FSMs, Equivalence, Regular Languages, Regular expressions, The memory required to recognize a language, Distinguishing one string from another, unions, Intersections and Complements, Finite automata, NFA, NFA with null transitions, Equivalence, Criterion for Regularity, Minimal Finite Automata, The pumping lemma, decision problems, Finite automata, Non-determinism and Kleen's Theorem, Regular and Non-regular languages.

2. Context – Free Grammars, Definition of CFG, example of familiar languages, unions, concatenations and closures of CFLs, Derivation Tree, Ambiguity, unambiguous CFG for algebraic expressions, Simplified forms and normal forms, Push down automata, definition, deterministic PDA, PDA to CFG and Vice Versa, Context Free and Non Context Free Languages, Pumping lemma for CFG, Intersection and complements of CFL.

3. Turing Machine Definition, Turing Machining as Language acceptors, combining TM, computing Partial Function with TM. Recursively Enumerable and Recursive Languages, Regular Grammars, context Sensitive grammars, Chomsky Hierarchy. Concept of insolvability & reducibility, Halting Problem, Post correspondence Problem, Rice theorem, P and NP. Polynomial-Time reductions and NP-Completeness.

Reference Books

1. John C. Martin: Introduction to Languages and the Theory of Computation, MGH.
2. Lewis & Papadimitriou: Elements of the Theory of Computation, PHI.
3. Daniel I.A. Cohen: Introduction to Computer Theory: John Wiley.
4. J.E. Hopcroft and J.D. Ullman: Introduction to Automata Theory Languages and Computation, Narosa.

Course Outcomes

1. To understand the foundations finite automata (deterministic or non-deterministic), FSM and master regular languages.
2. Be exposed to types of languages (type-0, type-1, type-2, type-3), context-free languages (type-2) and using pushdown automata (PDA) as its acceptor.
3. To design the Turing machines for various mathematical function, string function etc
4. To demonstrate concept of decidability, unsolvability and recursive.

Programme Electives – III and IV

Course Code	:	**CSPE31**
Course Title	:	**Software Verification and Validation**
Number of Credits	:	**4**
Prerequisites	:	**CSPC27**
Course Type	:	**PE**

Course Learning Objectives:

The objectives of the course is to prepare students for learning different techniques of verification and validation of object oriented and web based software.

Course Content:

1. Introduction to Testing & Software Verification [1]

Review of Functional and Structural Testing Techniques- Slice Based testing and Mutation Based testing. Verification methods, SRS document verification, SDD document verification, source code reviews, User document verification, Software project Audit

2. **Test Case Generation & Regression Testing [1]**
   Introduction to Use case diagram and use cases, test case generation from Use cases, validity checking and its guidelines, Data validation Strategies, Database Testing.
   Introduction to regression testing, regression test cases selection methods and their management, risk analysis, code coverage prioritization techniques

3. **Object Oriented Testing [1] [2]**
   Introduction to object orientation and testing [1], issues in object oriented testing [2], path testing, state based testing, class testing, integration testing [2], GUI testing [2].

4. **Testing metrics, models & Web Application Testing**
   Introduction to software metrics, categorization of metrics, object oriented metrics used in testing, elements to measure during testing, software quality attributes prediction models.
   Introduction to web testing, functional testing, user interface testing, usability testing, configuration and compatibility testing, security testing, performance testing, database testing, post deployment testing, web metrics.

Reference Books:

1. Yogesh Singh, Software Testing, Cambridge Publication, 2013.
2. Paul C. Jorgensen, Software Testing- A Craftsman's Approach, Auerbach Publication, 3rd Edition, 2012.

Course Outcomes:

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

1. Describe techniques in the areas of software inspection, software testing and software validation.
2. Apply different techniques for verification of the software.
3. Generate test cases for validation object oriented and web based software.

Course Code	**:**	**CSPE33**
Course Title	**:**	**Embedded System Development**
Number of	**:**	**4**
Prerequisites	**:**	**CSPC10, CSPC22, CSPC29**
Course Type	**:**	**PE**

Course Learning Objectives:

1. Present overview structure of embedded system with pipelining.
2. Understand the programming languages for embedded system design.
3. Develop an understanding of implementation methodology of Arduino and Rasberry Pi.

Course Content:

1. **Intel 8051 Micro controller:** Basic differences between Microprocessors and Microcontroller. Introduction to Intel 8051 Microcontroller, architecture, registers, Internal and External Memory. Instruction Set. On Chip Counters / Timers, Serial I/O, Interrupts and their use. Assembly language programming.

2. **Atmel And PIC Microcontrollers:** Introduction to Atmel and PIC C6X microcontrollers, architecture, registers, Internal and External Memory, Instruction Set, On Chip Counters / Timers, Serial I/O, Interrupts and their use. PWM, Watch dog Timer, ISP, IAP features. Assembly language programming.

3. **ARM7TDMI (Advanced RISC Machines):** ARM Architecture, Cortex-M3 Basics, Exceptions, Instruction Sets, NVIC, Interrupt Behaviour, Cortex-M3/M4 Programming, Exception Programming, Memory Protection Unit and other Cortex-M3 features, STM32L15xxx ARM Cortex M3/M4 Microcontroller Memory and Peripherals, Development & Debugging Tools.

4. **Open Source Embedded Development Board (Arduino):** Overview of open source embedded development board (Arduino), block diagram, pins of embedded development board, features of open source tool used for programming a development board, programming of embedded development board, Interface Serial Port with embedded development board, Program Raspberry Pi: a credit-card sized computer, Python programming for Raspberry Pi, Interacting and configuring the RPi OS, Porting of Linux Kernel and booting RPi.

   CASE STUDIES: Design of Embedded Systems using the Microcontroller– 8051/ Amtel/ Arduino/ Raspberry Pi, for applications in the area of Communications, Automotives, and industry.

Reference Books:

1. Raj Kumar, “Embedded Systems: Architecture, Programming and Design”, Tata McGraw Hill, Third Reprint, 2003.
2. John Catsoulis, O’Reilly, “Designing Embedded Hardware”, First Indian Reprint, 2003.
3. David E. Simon, “An Embedded Software Primer”, Pearson Education Asia, Fifth Indian Reprint, 2002.
4. Michael Barr, O’Reilly, “Programming Embedded Systems in C and C ++”, 1999.
5. J.W. Valvano, “Embedded Microcomputor System: Real Time Interfacing”, Brooks/Cole, 2000.
6. Jack Ganssle, “The Art of Designing Embedded Systems”, Newnes, 1999.
7. V.K. Madisetti, “VLSI Digital Signal Processing”, IEEE Press (NY, USA), 1995.
8. Arduino Cookbook, By Michael Margolis, O’Reilly , 2011.
9. Raspberry Pi Assembly Language RASPBIAN Beginners THIRD EDITION, CreateSpace Independent Publishing Platform, 19-Aug-2013.

Course outcomes:

1. Learn the concept of structure of embedded system.
2. Identify the needs of Arduino/Rasberry Pi in real life applications.

Course Code	**:**	**CSPE35**
Course Title	**:**	**Mobile Application Development**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	**CSIR12/CEIR12*, CSIR11, CSPC23**
Course Type	**:**	**PE**

Course Learning Objectives:

1. Introduction and characteristics of mobile applications.
2. Application models of mobile application frameworks.
3. Managing application data and User-interface design for mobile applications.
4. Integrating with cloud services.
5. Integrating networking, the OS and hardware into mobile-applications.
6. Addressing enterprise requirements in mobile applications – performance, scalability, modifiability, availability and security.
7. Testing methodologies for mobile applications**–** Publishing, deployment, maintenance and management.

Course content:

1. **Introduction to mobile devices:** Introduction to Mobile Computing, Introduction to Android Development Environment, Mobile devices vs. desktop devices, ARM and intel architectures, Power Management, Screen resolution, Touch interfaces, Application deployment, App Store, Google Play, Windows Store, Development environments: XCode, Eclipse, VS2012, PhoneGAP, etc.; Native vs. web applications. Factors in Developing Mobile Applications: Mobile Software Engineering, Frameworks and Tools, Generic UI Development, Android User; Graphics and Multimedia: Performance and Multithreading, Graphics and UI Performance, Android Graphics, Mobile Agents and Peer-to-Peer Architecture, Android Multimedia.

2. **Mobile OS Architectures:** Comparing and contrasting architectures of all three – Android, iOS and Windows, Underlying OS, Kernel structure and native level programming. Approaches to power management, Security. Android/iOS/Win 8 Survival and basic apps: Building a simple “Hello World” App in all three applications, App-structure, built-in Controls, file access, basic graphics. Platforms and Additional Issues: Development Process, Architecture, Design, Technology Selection, Mobile App Development Hurdles, Testing.

3. **Android/iOS/Win8 Apps:** DB access, network access, contacts/photos/etc. Underneath the frameworks: Native level programming on Android, Low-level programming on (jailbroken) iOS, Windows low level APIs. Intents and Services: Android Intents and Services, Characteristics of Mobile Applications, Successful Mobile Development; Storing and Retrieving Data: Synchronization and Replication of Mobile Data, Getting the Model Right, Android Storing and Retrieving Data, working with a Content Provider; Putting It All Together: Packaging and Deploying, Performance Best Practices, Android Field Service App.

4. **Advanced Topics:** Power Management: Wake locks and assertions, Low-level OS support, Writing power-smart applications. Augmented Reality via GPS and other sensors**:** GPS, Accelerometer, Camera. Mobile device security, in depth: Mobile malware, Device protections, iOS “Jailbreaking”, Android “rooting” and Windows’ “defenestration”; Security and Hacking: Active Transactions, More on Security, Hacking Android.

Reference Books:

1. Bill Phillips, Chris Stewart, Brian Hardy, and Kristin Marsicano, Android Programming: The Big Nerd Ranch Guide, Big Nerd Ranch LLC, 2nd edition, 2015.
2. Valentino Lee, Heather Schneider, and Robbie Schell, Mobile Applications: Architecture, Design and Development, Prentice Hall, 2004.
3. Tomasz Nurkiewicz and Ben Christensen, Reactive Programming with RxJava, O’Reilly Media, 2016.
4. Brian Fling, Mobile Design and Development, O’Reilly Media, Inc., 2009.
5. Maximiliano Firtman, Programming the Mobile Web, O’Reilly Media, Inc., 2nd ed., 2013.
6. Cristian Crumlish and Erin Malone, Designing Social Interfaces, 2nd ed., O’Reilly Media, Inc., 2014.
7. Suzanne Ginsburg, Designing the iPhone User Experience: A User-Centered Approach to Sketching and Prototyping iPhone Apps, Addison-Wesley Professional, 2010.

Course outcomes:

1. Understand technology and business trends impacting mobile applications
2. Be competent with the characterization and architecture of mobile applications.
3. Understand enterprise scale requirements of mobile applications.
4. Design and develop mobile applications using one application development framework.

Course Code	**:**	**CSPE37**
Course Title	**:**	**Complex Networks**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSPC26**
Course Type	**:**	**PE1**

Course Learning Objectives:

1. Conceptual introduction to the modern theory and applications of complex networks.
2. To develop the mathematical sophistication needed to understand and apply properties and methodologies to real life complex networks

Course content:

1. Ad hoc wireless networks, Challenges in wireless sensor networks, clustering algorithms, MST base, Lowest-ID based, and Dominating-set based clustering, Ad hoc routing in MANET and Sensor Networks, Time Synchronization, Localization, Target tracking, and other applications.

2. Peer-to-peer overlay networks, 1st and 2nd generation P2P networks, Koorde and De Bruijn Graph, Bittorrent and free riding. Analysis of Complex Networks, Vertex degrees, communities, Matching Index, centrality, Network motifs, Random networks, Small world networks, Scale-free networks,

3. Distance, Centrality and Prestige, Relationships, Special subgraphs, Data Clustering, Graph based clustering, Network motif discovery, Structural balance, cohesive subgroups, affiliation networks, Structural, Automorphic, and Regular Equivalence, Community detection algorithms, Social networks, Opportunistic Mobile Social Networks, Architecture, Community Detection, and Middleware, Routing, Multicast in opportunistic networks, Information Diffusion, Influence maximization, Characterizing link connectivity, Data dissemination, Privacy preserving, Encounter based opportunistic social discovery.

Books

1. Stefano Basagni, Marco Conti, Silvia Giordano, and Ivan Stojmenovic, Mobile Ad Hoc Networking: Cutting Edge Directions, 2/e Wiley 2013**.**
2. Maarten van Steen, Graph Theory and Complex Networks: An Introduction, 2010.
3. KayhanErciyes, Complex Networks: An Algorithmic Perspective, CRC Press 2015.
4. Jie Wu and Yunsheng Wang, Opportunistic Mobile Social Networks, CRC Press 2015.

Course Outcomes:

1. Able to explain the main concepts of complex network analysis.
2. Able to identify network models and explain their structures.
3. Explain emergent features of complex networks with network models.
4. Analyse and design network models.
5. Familiarity with the new research in this evolving area.

Course Code	:	**CSPE71**
Course Title	:	**Advance Database Systems**
Number of Credits	:	**4**
Prerequisites(Course Code)	:	**CSPC25**
Course Type	:	**PE**

Course Learning Objectives

1. To understand the application and fundamentals of Advance Database Systems in real- world.
2. To study the basics of query processing and optimization techniques in Advanced Database systems.
3. To lean the basic issues of transaction processing and concurrency control in Distributed and Multidatabase system.
4. To develop the knowledge of various design and implementation issues of Advance Database for given business requirement.

Course Content

1. **Introduction**: Overview of Distributed Database, Data Delivery Alternatives, Promises of DDBSs, Distributed DBMS Architecture, Design Issues, Fragmentation: Horizontal, Vertical, Hybrid Fragmentation, Allocation and Information Requirements, Allocation Model, DD Design approach: Top-down and Bottom-Up Design Methodology, Schema Matching, Schema Heterogeneity, Linguistic Matching Approaches, Constraint-based Matching Approaches, Learning-based Matching, Combined Matching Approaches, Schema Integration, Schema Mapping.

2. **Distributed Database Query Processing**: Overview of Distributed Query Processing, Complexity of Relational Algebra Operations, Layers of Query Processing, Query Decomposition: Normalization, Analysis, Elimination of Redundancy, Rewriting, Localization of Distributed Data, Reduction for fragmentation, Query Optimization: Static, Dynamic and Hybrid,

3. **Multidatabase Query Processing**: Issues in Multidatabase Query Processing , Multidatabase Query Processing Architecture, Query Rewriting Using Views, Datalog Terminology, Rewriting in GAV,

Rewriting in LAV, Query Optimization and Execution, Heterogeneous Cost Modeling, Heterogeneous Query Optimization, Adaptive Query Processing , Query Translation and Execution.

4. **Introduction to Transaction Management:** Definition of a Transaction, Termination Conditions of Transactions, Characterization of Transactions, Formalization of the Transaction Concept, Properties of Transactions, ACID, Types of Transactions, Flat Transactions, Nested Transactions, Workflows.

Reference Books

1. M. Tamer Özsu, Patrick Valduriez, Principles of Distributed Database Systems, 3rd edition, Springer, 2012.
2. Hector Garcia-Molina, Jeffrey Ullman,, Jennifer Widom, Database System Implementation, Prentice-Hall, 2000.
3. A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts,. 5th Edition. McGraw-Hill.

Course Outcomes

1. To understand fundamental concepts, principles and applications of Adv. Database Systems.
2. To demonstrate the knowledge of database system and evaluate Database design.
3. To apply state-of-art techniques to design advanced techniques in Database Systems.
4. Comprehend contemporary trend and issues in Advance Database System design related to research domain, industry and government.

Course Code	**:**	**CSPE73**
Course Title	**:**	**Service Oriented Architecture (SOA)**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSPC27**
Course Type	**:**	**PE**

Course Learning Objectives

1. To learn service oriented analysis techniques.
2. To learn technology underlying the service design.
3. To learn advanced concepts such as service composition, orchestration and Choreography.
4. To know about various Web Service specification standards.

Course Content

1. **Introduction**: Fundamentals of Service Oriented Architecture, Characteristics of SOA, Comparison of SOA with Client-Server and Distributed Internet Architectures, Anatomy of SOA, Principles of Service Orientation

2. **Service Oriented Architecture in Web Services**: Fundamentals of Web services, Service descriptions, Message exchange Patterns, Coordination, Atomic Transactions, Business activities: Orchestration and Choreography, Service layer abstraction, Application Service Layer, Business Service Layer, Orchestration Service Layer

3. **Design SOA**: Service oriented analysis, Business-centric SOA, Deriving business services service modelling, Service Oriented Design, WSDL basics, SOAP basics, SOA composition guidelines, Entity-centric business service design, Application service design, Task centric business service design, BPEL overview: Process, Elements and functions, Web Service Coordination overview: elements, Web service business activity & atomic transaction coordination type, Business process design, Web Service Choreography, Web Service Policy elements, Web Service Security, XML Signature element

4. **SOA Platforms**: SOA platform basics, SOA support in J2EE, Java API for XML based web services (JAX-WS), Java architecture for XML binding (JAXB), Java API for XML Registries (JAXR), Java API for XML based RPC (JAX-RPC), Web Services Interoperability Technologies (WSIT), SOA support in

.NET, Common Language Runtime, ASP.NET web forms, ASP.NET web services, Web Services Enhancements (WSE)

Reference Books

1. Thomas Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Pearson Education, 2009.
2. Thomas Erl, SOA Principles of Service Design, The Prentice Hall, 2005.
3. Newcomer, Lomow, Understanding SOA with Web Services, Pearson Education, 2005.
4. Sandeep Chatterjee, James Webber, Developing Enterprise Web Services: An Architect's Guide, Pearson Education, 2005.

Course Outcomes

1. To understand major software architectural styles, design patterns, and frameworks.
2. To identify and assess the quality attributes of a system at the architectural level.
3. To create architectural alternatives for real-world problems and evaluate system's architecture.
4. To comprehend the contemporary trends and issues related to Software Architectures in research and industry.

Course Code	**:**	**CSPE75**
Course Title	**:**	**Advanced Computer Architecture**
Number of Credits	**:**	**4**
Prerequisites	**:**	**CSPC10, CSPC22**
Course Type	**:**	**PE**

Course Learning Objectives:

The objective of this course is to give students a clear understanding of the modern computer architecture. Students will learn the design and analysis of high performance (multi)processors and supporting subsystems from the quantitative aspect.

Course content

1. Introduction to Parallel Processing

Parallelism in uni-processor system, pipelining, basic assumptions, design techniques, designing pipelined data path, propagating an instruction queue through the pipeline, pipeline hazards and their detection, forwarding, instruction level parallelism, super-scalar architecture, general pipeline and reservation tables, principles of linear pipelining, Instruction and arithmetic pipeline, principlesof designing pipelined processor, micro-programming, general approach, micro-code engine, control store, branching and looping, horizontal and vertical micro-code scheme with example, alternative approach to micro-coding.

2. Issues in the Architecture of Distributed Systems

(a) Introduction, Examples, Distributed Systems versus Parallel Systems, Partial Orders, Models of Distributed Systems, Architectural Models, Fundamental Models, Interleaving Model, Happened Before Model, Potential Causality Model, Appropriate Model, Models Based on states, Deposet.

(b) Program correctness, Correctness criteria, Safety and Liveness properties, Correctness proofs, Predicate logic, Assertional reasoning, Well-Founded sets, Predicate Transformers.

(c) Concept of Logical Time, Logical Clock and its implementation, Limitations of Lamport's Logical Clock, Vector Clock and its implementation, Matrix Clock, and other optimal clocks, Using Induction to prove properties of Clocks, Verifying Clock Algorithms, Events, Process States, Clock Skew and Clock Drift, Synchronizing Physical Clocks, Cristian's Method, The Berkeley Algorithm, The Network Time Protocol, Distributed Debugging.

(d) Synchronization Problems, Mutual exclusion, Shared memory algorithms, Peterson andLamport's Bakery Algorithm, Token passing algorithms, Suzuki-Kasami algorithm, Raymond's algorithm, Message passing algorithms, Lamport's algorithm, Ricart-Agrawala algorithm, Maekawa algorithm, Leader election, Garci-Molina Bully algorithm, Chang-Roberts unidirectional ring algorithm, Hirschberg-Sinclair bidirectional algorithm, Global States and Consistent Cuts, Chandi-

Lamport and Lai-Yang Snapshot Algorithm, Global State Collection, Dijkstra-Scholten and other termination algorithms, Chandi-Misra-Hass and other deadlock detection algorithms.

(e) Fault tolerance, Agreement problems, Asynchronous Distributed Consensus, Impossibility(FLP) result and its proof, Consensus in Synchronous systems, Byzantine Generals Problem (BGP), Consensus with oral and signed messages, Failure detectors.

(f) Naming (Tanenbaum-Steen, Ch. 4), Transactions and Concurrency control (CDK, Ch. 12), Distributed Transaction (CDK, Ch. 13), Replication (CDK, Ch. 14)

BOOKS

1. Hwang and F.A.Briggs: Computer Architecture and Parallel Processing, McGraw Hill.
2. Distributed Systems: Concepts and Design; G Colouris, J Dollimore, T Kindberg 3/e Pearson Ed. 2002. (CDK)
3. Distributed Systems: Principles and Paradigm; Andrew S Tanenbaum, Maarten van Steen 3/e Pearson Ed. 2002.
4. Elements of Distributed Computing, VK Garg, Wiley-Interscience Publishers, 2002.

Course Outcomes

1. Understand the modern concepts of computer architecture.
2. Investigate modern design structures of Pipelined and Multiprocessors systems.
3. Design new multiprocessors systems.

Course Code	**:**	**CSPE77**
Course Title	**:**	**Image processing**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	**MAIR12**
Course Type	**:**	**PE**

Course Learning Objectives:

Image Processing is a major course in the computer science major, designed to teach students the fundamentals of digital image processing on the computer. The primary objective of this course is to introduce students to basic principles of digital images, image data structures, and image processing algorithms.

Course Content:

1. Introduction

Image Processing Fourier Transform and Z-Transform, Causality and stability, Toeplit and Circulate Matrics, orthogonal and unitary Matrices and Kroenker product, Markov Processes, KI Transform, Mean square Estimates and Orthogonal Principles

Image sampling & Quantization: Band Limited Image, Sampling Versus Replication, Reconstruction of Image from samples Sampling Theorem, Sampling Theorem for Random Fields, Optimal Sampling, Non-rectangular Grid Sampling, Sampling Aperture, Display Aperture/ Interpolation Functions, Lagrange Interpolation, Moire Effect

Image Quantization: Uniform Optimal Quantizer, Properties of Mean Square Quantizer, Compandor Design, Visual Quantization

2. Image Transforms

Two Dimensional Orthogonal and Unitary Transforms and their properties, One Dimensional and Two Dimensional DFT, Cosine, Sine, Hadamard, SLANT, HAAR and KI Transforms and their properties, Approximation to KI Transforms.

Image representation by stochastic models: One Dimensional Causal Models, AR, ARMA models, Spectral factorization, Non Causal Representation, Image Decomposition.

3. Image Enhancement and Restoration

Point Operation, Histogram Modeling, Spatial Operations, Transform Operations, Multispectral Image Enhancement.

Image Filtering: Image Observation Models, Inverse and Wiener filter, FIR Wiener Filters, Filtering using Image Transform, Causal Models and recursive filtering, Maximum entropy restoration, Extrapolation of band limited signal.

4. **Image Analysis and Image Compression**

   Spatial feature extraction, Edge detection, Boundary extraction, Boundary representations, Region representations, Moment representations, Structures, Texture, Image Segmentation, Image Classification

   Image Reconstruction from Projections:

   Data Compression: Pixel Coding, Predictive Techniques, Transform Coding Theory, Transform coding of Image, Coding of two-tone image.

Reference Books:

1. Anil Jain, Fundamentals of Digital Image Processing, Pearson, 1988.
2. Richard E Woods and Rafael C Gonzales, Digital Image Processing, Pearson, 3rd edition, 2009.
3. William. K. Pratt, Digital Image Processing, John Wiley & Sons, 4th edition, 2007.

Course outcomes:

1. Acquire the fundamental concepts of a digital image processing system
2. Apply knowledge of various mathematical tools used for 1D and 2D signal analysis and processing
3. Analyze 2D signals in the spatial and frequency domain.
4. Design and implement algorithms for digital image processing operations.
5. Ability to apply image processing techniques to solve various real time problems.

CSIR30 / CSIR32 – Internship / Project Work

Course Code	**:**	**CSPC41**
Course Title	**:**	**Machine Learning**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	**CSPC33, MAIR13, MAIR12**
Course Type	**:**	**PC**

Course Learning Objectives:

1. Be able to understand and apply supervised and unsupervised learning algorithms.
2. Understand the fundamental concepts in machine learning and popular machine learning algorithms.
3. Understand the concept of Deep Learning
4. Be able to solve problems related to the application of machine learning algorithms with programming in Python/ MatLab.

Course Contents:

1. **Introduction**

   Basic definitions, types of learning, hypothesis space and inductive bias, evaluation, cross-validation, Instance based learning, Feature reduction, Collaborative filtering based recommendation, Gradient Decent learning.

2. **Kernel Methods and Support Vector Machines**

   The Two-Class Problem, Dual Representation, Soft Margin Classification; Origins of Kernel methods, Kernel Mapping, The Kernel Trick; Constructing Kernels, Support Vector Machines: Formulation and Computation; Radial Basis Function Networks; Positive Semi-Definite Kernels, Linear Kernel, Polynomial Kernel.

3. **Clustering and Learning**

Basic Clustering Techniques, Standard k-Means (Lloyd) Algorithm, Generalized Clustering, Overpartitioning, Merging, Modifications to the k-Means Algorithm, k-Means Wrappers, Rough k-Means, Fuzzy k-Means, k-Harmonic Means Algorithm, Hybrid Clustering Algorithms; Estimation using Incomplete Data, Two classes, Multiple classes, Least squares for classification, Fisher's linear discriminant, Relation to least squares, Fisher's discriminant for multiple classes

4. **Deep Learning**
   Neural network: Perceptron, multilayer network, backpropagation, Introduction to deep neural network: Convolutional Neural Networks, its Architectures, convolution layers, pooling strategies and softmax layer,

Reference Books:

1. T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning, 2e, 2008.
2. Christopher Bishop. Pattern Recognition and Machine Learning. 2e.
3. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
4. Introduction to Machine Learning Edition 2, by Ethem Alpaydin

Course Outcome:

1. Recognize major programming languages.
2. Explain the stages of programming language interpretation and compilation.
3. Understand fundamental concepts in machine learning and popular machine learning algorithms.
4. Explain basics of computational learning theory.
5. Hands-on problem solving with programming in Python and MatLab

Course Code	**:**	**CSPC43**
Course Title	**:**	**Distributed computing**
Number of Credits	**:**	**3**
Prerequisites(Course Code)	**:**	
Course Type	**:**	**PC**

Course Learning Objectives:

1. To understand fundamental concepts of distributed computing.
2. To acquire knowledge to design and develop protocols for distributed systems and networks.

Course content

1. Fundamental issues in Distributed Systems, Models of Distributed Computations, Handling time through clocks, Physical Time and Hardware Clock Synchronization, Logical Time and Software Clocks, Message Ordering Protocols, Interprocess Communication, Shared Memory Vs Message Passing Models, Distributed and Localized Algorithms, Evaluation of Correctness and Performance.

2. Protocols for Routing and Designing Virtual Constructs, MST, MIS, and CDS, Distributed Mutual Exclusion, Leader Election,

3. Distributed Deadlock Detection, Termination Detection, Global State Computation and Snapshot Recording.

4. Global Predicate Detection, Fault-Tolerance Issues, Basic Techniques for handling Crash Failure in Distributed Systems, Checkpointing and Rollback Recovery, Z-path and Z-cycles, Other Types of Failures in Distributed Computing Systems, Consensus and Agreement Protocols, FLP impossibility, Byzantine Generals Problem, Failure Detectors, Wait Freedom, Self-Stabilization, Peer-to-Peer Computing, Overlay Structures, Distributed Scheduling and Load Balancing, Naming, Security.

Text Books, and/or reference material

1. A D Kshemkalyani and MukeshSinghal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press 2008.
2. G Colouris, J Dollimore, and T Kindberg, Distributed Systems: Concepts and Design, 3/e Pearson Ed. 2002.
3. A S Tanenbaum and M van Steen, Distributed Systems: Principles and Paradigms, 3/e Pearson Ed. 2002.
4. V K Garg, Principles of Distributed Systems, Kluwer Academic Publishers, 1996.
5. Sukumar Ghosh, Distributed Systems: An Algorithmic Approach, 2/e Chapman& Hall.
6. V K Garg, Principles of Distributed Computing, IEEE Press.
7. Nancy Lynch, Distributed Algorithms, Morgan Kaufmann Press.
   Gerard Tel, Introduction to Distributed Algorithms, Cambridge University Press.

Course Outcome

1. Able to apply knowledge of operating systems and networks to distributed systems
2. Design distributed algorithms for complex computing problems
3. Analyze the performance of algorithms and evaluate the scalability of solutions
4. Adapt existing centralized algorithms to distributed set up
5. Simulate the algorithms in distributed environment

Course Code	**:**	**CSPC45**
Course Title	**:**	**Information Security**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	**MAIR13, CSPC26**
Course Type	**:**	**PC**

Course Learning Objectives:

1. To introduce the concept of information security.
2. To make students familiar with different aspects of access control, system security and its applications.
3. To design security solutions for different operation systems and databases
4. To discuss and analyze various legal issues and cyber laws.

Course Content:

1. **Introduction**
   Meaning of security, privacy, attacks, vulnerabilities and protection, CIA Triad, threat, control, layers of information security, Computer Criminals, Methods of defense, Cryptography, Steganography, ciphers, encryption, decryption, Key exchange, Uses of Encryption, digital signature.

2. **Access Control and Legal issues**
   The principle of least privilege, Authentication, Authorization, Access control system operation, Access control system components, Access control topology, Access control mechanisms, mandatory access control, Discretionary access control, Role based access control, Attribute-based access control, Rule-based access control, Identity-based access control, Legal issues, Ethical issues, Privacy, Cyber Crime, Need for Cyber Law, Cyber laws, Intellectual property issues in Cyber space.

3. **System and Operating Security**
   Overview, Security policy, File Protection Mechanisms, User Authentication, Security models, Trusted Operating System Design, Host based Intrusion detection system, Network based Intrusion detection system, Host based Intrusion prevention system, Network based Intrusion prevention system.

4. **Network and Database Security**
   Network security Concepts, Network threats, Security controls, Firewalls, Protecting Programs and Data, Secure Program, sniffing, spoofing, Non Malicious Program errors, Malware, Viruses and other malicious code, Control against Program, Honeypot, Database security requirements, privacy and Integrity, Sensitive Data, Multilevel database.

Assignment (Implementation of any Security algorithm from above related topics, as an assignment)

Suggested Books:

1. Charles P. Pfleeger, Share Lawrence Pfleeger, Security in Computing, Pearson Education, 2/e.
2. William Stallings, Cryptography and Network Security, PHI, 4/e
3. Neal Koblitz, A Course in Number Theory and Cryptography, Springer 2006.

Reference Books:

1. An Introduction to theory of numbers, Niven, Zuckerman and Montgomery, (Wiley 2006)
2. Elliptic curves: number theory and cryptography, Lawrence C. Washington, (Chapman & Hall/CRC 2003)
3. Charlie Kaufman, Perlman & S Peeciner, Network Security, Pearson Education, 2/e.

Course Outcomes:

On successful completion of this, course students will:

1. Able to understand the information security concepts.
2. Able to analyses different aspects of access control, system security and its applications.
3. Able to design and develop security solutions for different operation systems and databases
4. Able to familiar with various legal issues and cyber laws.

Programme Elective – V

Course Code	**:**	**CSPE41**
Course Title	**:**	**Cloud and Grid Computing**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	**CSPC26**
Course Type	**:**	**PE**

Course Learning Objectives:

The objective of the course is to give students a comprehensive view of storage and networking infrastructures for highly virtualized cloud ready deployments. The course discusses the concepts and features related to Virtualized datacenter and cloud, Information storage, security and design of applications.

Course Content

1. Overview of Computing Paradigm, Recent trends in Computing. Distributed Computing, Cluster Computing, Grid Computing, Utility Computing.
2. Cloud Computing, Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers, Pros and Cons of Cloud Computing, Benefits of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing, Cloud Computing Architecture, Service Models (XaaS), Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS).
3. Deployment Models, Public cloud, Private cloud, Hybrid cloud, Community cloud, Dynamic provisioning and resource management,Cloud security,
4. Case Study on Open Source & Commercial Clouds: Eucalyptus, Microsoft Azure, Amazon EC2.

Reference Books:

1. Cloud Computing (Wind) by Kumar Saurabh, 2nd Edison, Wiley India.
2. Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley, 2011.

3. Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer, 2012.

Course outcomes

1. To understand and apply the knowledge of systems protocols and mechanisms to support Grid and Cloud computing
2. To analyze and implement the concepts of Cloud Computing
3. To deploy applications on cloud middleware
4. To understand and resolve security issues in various applications

Course Code	**:**	**CSPE43**
Course Title	**:**	**Natural Language Processing**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	**CSPE210**
Course Type	**:**	**PE**

Course Learning Objectives:

The objective of this course is to give students a clear understanding of linguistics methods, various tools and aspects of NLP like syntax and semantic analysis, parsing, machine translation, information retrieval and statistical discourse processing.

Course Content

1. **Introduction to Natural Language Processing (NLP):** Definition, History, Applications of NLP, Goals of NLP.

2. **Words and Phonetics**: Regular expressions and Automata, Morphology and phonetics fundamentals, morphological diversity of Indian languages, morphology paradigms, and finite state machine based morphology, Computational Phonology and Text-to-Speech, Probabilistic Models of Pronunciation and Spelling, N-grams, HMMs and Speech Recognition, Wordnet and linking.

3. **Parsing:** Part-of Speech Tagging, theories of parsing, syntactic and statistical parsing, parsing algorithms, hybrid of rule based and probabilistic parsing, scope ambiguity and attachment ambiguity resolution, Tree banks.

5. **Discourse and dialogue:** discourse and dialogue analysis, anaphora resolution, named entity resolution, event anaphora, Information extraction and retrieval.

6. **Applications:** sentiment analysis, text entailment, machine translation, automated speech recognition systems, question-answering based systems, shallow parsers.

Reference Books:

1. Jurafsky, D. & J. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing Computational Linguistics, and Speech Recognition" Prentice Hall, 2000.
2. Grosz, B.J., Sparck Jones, K. & Webber, B.L. (eds) "Readings in natural language processing", Los Altos, CA. Morgan Kaufmann, 1986.
3. Allen, J., "Natural Language Understanding", Redwood City, CA: 1994. Benjamin/Cummings.
4. Bharti, Akshar,Chaitanya Vineet, Sangal Rajeev, "Natural Language Processing", Prentice Hall.

Course Outcomes

1. Develop background of translation, information retrieval and statistical discourse processing.
2. Introduction to various recent statistical methods in natural language processing.
3. Develop familiarity to linguistics and their application to part-of-speech tagging.

Course Code	**:**	**CSPE45**
Course Title	**:**	**Computer Vision**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	**MAIR13, CSPC11**
Course Type	**:**	**PE**

Course Learning Objectives:

1. Be familiar with both the theoretical and practical aspects of computing with images;
2. Have described the foundation of image formation, measurement, and analysis;
3. Have implemented common methods for robust image matching and alignment;
4. Understand the geometric relationships between 2D images and the 3D world.
5. Have gained exposure to object and scene recognition and categorization from images;
6. Able to develop the practical skills necessary to build computer vision applications.

Course content

1. Digital Image Formation and low-level processing

Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.

2. Feature Extraction

Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT.

3. Shape Representation, Segmentation and Object Recognition

Shape Representation and Segmentation, Deformable curves and surfaces, Snakes and active contours, Level set representations, Fourier and Wavelet Descriptors, Medial Representations ,Multiresolution analysis, Hough transforms and other simple object recognition Methods, Shape Correspondence and Shape Matching, Shape priors for recognition.

4. Motion Estimation

Regularization Theory, Optical Computation, Stereo Vision, Motion Estimation, Structure from Motion.

Reference Books:

1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.
2. Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 2003.
3. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004.
4. K. Fukunaga; Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990.
5. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Addison- Wesley, 1992.

Journals

1. IEEE-T-PAMI (IEEE Transactions on Pattern Analysis and Machine Intelligence).
2. IJCV (International Journal of Computer Vision) - Springer.

Course Outcome:

1. Able to demonstrate knowledge and understanding of Human and computer vision systems.
2. Understand current approaches to image formation and image modeling.
3. Analyze and design a range of algorithms for image processing and computer vision
4. Develop and evaluate solutions to problems in computer vision

Course Code	**:**	**CSPE47**

Course Title	**:**	**Graphics & Visualization**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	**MAIR12**
Course Type	**:**	**PE**

Course Learning Objectives:

1. Comprehensive introduction for computer graphics leading to the ability to understand contemporary terminology, progress, issues, and trends
2. Involves processing and transforming large, incomprehensible datasets into a format more easily understood by humans
3. Introduce theoretical methods for two-dimensional and three-dimensional graphics with applications to visualization techniques

Course Content:

1. Introduction, Scan Conversion, 2D & 3D - Transformation and Viewing:

Overview of computer graphics, representing pictures, preparing pictures for presentation, presenting previously prepared pictures, CRT technologies - Raster Scan Display, Computer graphics software. Scan Conversion: Points & lines, Line drawing algorithms; DDA algorithm, Bresenham’s line algorithm, Circle generation algorithm; Ellipse generating algorithm; scan line polygon, fill algorithm, boundary fill algorithm, flood fill algorithm. 2D & 3D - Transformation and Viewing: Translation, rotation, scaling; Matrix representations & homogeneous coordinates, transformations between coordinate systems; reflection shear; Transformation of points, lines, parallel lines, intersecting lines; Viewing pipeline, Window to Viewport co-ordinate transformation, clipping operations, point clipping, line clipping, clipping circles, polygons & ellipse. 3D Transformation: Translation, rotation, scaling & other transformations, Rotation about an arbitrary axis in space, reflection through an arbitrary plane, general parallel projection transformation

2. Clipping, Curves Fractals, Geometric modeling, Hidden Surfaces and Color & Shading Models:

Viewport clipping, 3D viewing, perspectives & Depth Cueing. Curves and Fractals: Curve representation, surfaces, designs, Bezier curves, B-spline curves, end conditions for periodic B-spline curves, rational B-spline curves. Geometric modeling: Elementary geometric algorithms for polygons, boundary representations, constructive solid geometry, spatial data structures. Hidden Surfaces and Color & Shading Models: Depth comparison, Z-buffer algorithm, Back face detection, BSP tree method, the Printer’s algorithm, scan-line algorithm; Hidden line elimination, wire frame methods, fractal – geometry. Color & Shading Models: Introduction, Modeling Light Intensities and Sources, Diffuse Reflection, Lambert’s Cosine Law, Specular Reflection, Halftoning, Color Models - RGB Color, CMY Color, realistic image synthesis techniques, textures and image-based rendering

3. Visual saliency, perceptual attractiveness, Visualization of experimental and simulated data:

Perception-Oriented and Non-Photorealistic Rendering, human perception of saliency on object surfaces, relief mapping of textures. Visualization of experimental and simulated data: Surface construction from scattered data, 3-D data arrays and 2-D cross sections, Elevation maps, topological maps, contour maps and intensity maps; fractals for visualization of complex and large data sets, Algebraic stochastic and Geometric fractals, Modeling of natural forms and textures using fractals, Visualization of multi variate relations, Flow visualization and hyper streamlines, Visualization of Meteorological, cosmological, seismic, biological data for scientific decision making, Data and Geometry, Height field and contours, Visualizing surfaces and scalar fields, Isosurfaces and marching cubes, Direct volume rendering, Vector-field visualization, Tensor-visualization

4. Animation, Visual Realism, Image/Video based rendering, Animation and Capture:

Modelling issues in dynamic visualization, Behavioural animation, walk through coordinate transformation and view transformation, virtual reality interfaces, Interactive and immersive systems for Prototyping and visualization, Visualization in concurrent engineering, Interactive multimedia technology and standards for Video-Graphics-Audio integration and tele-video conferencing. Visual Realism: Natural effects for object rendering, iterative ray tracing, radiosity, plenoptic function, Perlin noise, lighting & shadow, materials effect, BRDF, volumes and photons, fire and water, fog, smoke modeling etc. Image/Video based rendering: Video mosaic synthesis, video resizing and stabilization, Novel view Synthesis; Photosynth, retargeting, 3D cut and paste. Animation and capture: Motion Control, Orientation Parameterization, Quaternions, Animating Articulated Structures, Inverse Kinematics, hard and soft constraints, implicit surfaces; Stylized rendering and illusion, facial animation, appearance capture.

Reference Books:

1. Edward Angel and Dave Shreiner, Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL, Pearson, 6th edition, 2011.
2. F.S. Hill Jr., Computer Graphics Using OpenGL, Pearson, 3rd edition, 2006.
3. John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, and Kurt Akeley, Computer Graphics: Principles and Practice, Pearson Education, 1st edition, 2013.

Course outcomes:

1. Use the underlying architecture, algorithms, mathematical concepts, supporting computer graphics
2. Enhance their perspective of modern computer system with modeling, analysis and interpretation of 2D and 3D
3. Fundamental graphic design theory and principles as they relate to current trends in visual communication
4. Organize information to make compelling and experimental visual expressions for presentation
5. Explain and understand fundamental concepts within information visualization and scientific visualization

Course Code	**:**	**CSPC40**
Course Title	**:**	**Big Data Analytics**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	**CSPC214**
Course Type	**:**	**PC**

Course Learning Objectives

1. To understand the concept and challenge of big data and why traditional technology is inadequate to analyze the big data.
2. To collect, manage, store, query, and analyze various types of big data.
3. Gain hands-on experience on large-scale analytics tools to solve big data problems.
4. To study the impact of big data analysis for societal and business decisions.

Course Content

1. **Introduction:** Overviews of Big Data, State of the Practice in Analytics, The Data Scientist, Big Data Analytics in Industry Verticals, Data Analytics Lifecycle Challenges of Conventional Systems, Statistical Concepts: Sampling Distributions, Re-Sampling, Statistical Inference, Prediction Error, Regression Modelling , Multivariate Analysis , Bayesian Modelling .

2. **Mining Data Streams**: 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 Prediction.

3. **Frequent Itemset 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 based Techniques: Hierarchical, K-Means etc., Clustering High Dimensional Data, CLIQUE And PROCLUS, Frequent Pattern based Clustering Methods, Clustering in Non-Euclidean Space, Clustering for Streams and Parallelism.

4. **Frameworks and Visualization**: Overview of MapReduce, Hadoop, Hive, MapR, Sharding, NoSQL Databases, S3, HADOOP, Distributed File System (HDFS), Visualizations: Visual Data Analysis Techniques , Interaction Technique and Applications.

Reference Books

1. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.
2. A. Rajaraman, J.D. Ullman, Mining of Massive Datasets, Cambridge University Press, 2012.
3. Bill Franks, Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, John Wiley & sons, 2012.
4. Glenn J. Myatt, Making Sense of Data, John Wiley & Sons, 2007
5. Pete Warden, Big Data Glossary, O'Reilly, 2011.
6. J. Han, M. Kamber , Data Mining Concepts and Techniques, 2nd Edition, Elsevier, Reprinted 2008.

Course Outcomes

1. To become proficient in recognizing challenges faced by applications dealing with very large data as well as in proposing scalable solutions.
2. To design efficient algorithms for mining the data databases.
3. To model a framework for visualization of big data analytics for business user.
4. To understand the significance of Big Data Analysis in business intelligence, scientific discovery, and day-to-day life.

Programme Elective – VI

Course Code	**:**	**CSPE40**
Course Title	**:**	**High Performance Computing**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	**CSPE315**
Course Type	**:**	**PE**

Course Learning Objectives:

The objective of course, is to provide a comprehensive introduction to parallel algorithms and parallel programming, with strong emphasis on the design of parallel algorithms and their rigorous analysis. Exposure to parallel programming is provided through programming assignments using MPI. Throughout the course, the design of algorithms is interlaced with the programming techniques useful in coding them.

Course Content:

1. Performance measures: Speedup, efficiency and scalability. Model of parallel computation and basic communication primitives. Parallel prefix and applications, Parallel sorting, Embeddings, Parallel matrix algorithms.

2. Communication networks for parallel computers and parallel models of computation, Parallel fast Fourier transforms.

3. Parallel Programming with MPI: Writing and executing MPI programs, collective communication, grouping data for communication, communicators and topologies.

4. Parallel random number generation, Parallel Octrees, Parallel N-body methods, Parallel Bayesian network construction.

Suggested Books:

1. Introduction to Parallel Computing: Design and Analysis of Algorithms by A. Grama, A. Gupta, G. Karypis and V. Kumar.
2. Parallel Programming with MPI by Peter S. Pacheco.
3. Introduction to High Performance Scientific Computing Victor Eijkhout, Edmond Chow, Robert van de Geijn.

Course Outcomes

1. Understand in a general sense the architecture of high performance computers.
2. Understand how the architecture of high performance computers affects the speed of programs run on HPCs.
3. Understand how memory access affects the speed of HPC programs.
4. Understand Amdahl's law for parallel and serial computing.
5. Understand the importance of communication overhead in high performance computing.
6. Understand some of the general types of parallel computers.
7. Understand how different types of problems are best suited for different types of parallel computers.
8. Understand some of the practical aspects of message passing on MIMD machines.

Course Code	**:**	**CSPE42**
Course Title	**:**	**Human Computer Interaction**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	
Course Type	**:**	**PE**

Course Learning Objectives:

1. Describe and apply core theories, models and methodologies from the field of HCI
2. Implement simple graphical user interfaces
3. Discuss current research in the field of HCI

Course Content:

1. Introduction, Task-centred system design, User-centred design and prototyping:

Human-Computer Interaction. Task-centred system design: Task-centered process, development of task examples, evaluation of designs through a task-centered walk-through. User-centred design and prototyping: Assumptions, participatory design, methods for involving the user, prototyping, low fidelity prototypes, medium fidelity

2. Methods for evaluation of interfaces with users and Psychology of everyday things:

Goals of evaluation, approaches, ethics, introspection, extracting the conceptual model, direct observation, constructive interaction, interviews and questionnaires, continuous evaluation via user feedback and field studies, choosing an evaluation method. Psychology of everyday things: Psychopathology of everyday things, examples, concepts for designing everyday things. Beyond screen design: characteristics of good representations, information visualization, Tufte's guidelines, visual variables, metaphors, direct manipulation

3. Graphical screen design, Design principles and usability heuristics:

Graphical design concepts, components of visible language, graphical design by grids. Design principles and usability heuristics: Design principles, principles to support usability, golden rules and heuristics, HCI patterns

4. HCI design standards, Past and future of HCI:

Process-oriented standards, product-oriented standards, strengths and limitations of HCI Standards. Past and future of HCI: The past, present and future, perceptual interfaces, context-awareness and perception

Reference Books:

1. Dix A., Finlay J., Abowd G. D. and Beale R., Human Computer Interaction, Pearson Education, 3rd edition, 2005.
2. Preece J., Rogers Y., Sharp H., Baniyon D., Holland S. and Carey T. Human Computer Interaction, Addison Wesley, 1st edition, 1994.

Course outcomes:

1. Understand the basics of human and computational abilities and limitations

2. Understand basic theories, tools and techniques in HCI
3. Understand the fundamental aspects of designing and evaluating interfaces
4. Practice a variety of simple methods for evaluating the quality of a user interface
5. Apply appropriate HCI techniques to design systems that are usable by people

Course Code	**:**	**CSPE44**
Course Title	**:**	**Compiler Design**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	**CSPC37**
Course Type	**:**	**PE**

Course Learning Objectives:

The objective of this course is to give students a clear understanding of the major concept areas of language translation and to enrich the knowledge in various phases of compiler and its use, parsing techniques, code optimization techniques, machine code generation, and use of symbol table.

Course Content

1. Introduction and types of translators, the structure of a compiler, design of lexical analyzer,specification & implementation of lexical analyzer. Parsers, shift-reduce parsing, operator-precedence parsing, top-down parsing, recursive descent parsing, predictive parsers.

2. LR Parsers, the canonical collection of L R (O) items, construction of SLR parsing tables, constructing canonical L.R. Parsing tables, Constructing LALR parsing tables, implementation of L R Parsing tables Syntax-directed translation schemes, implementation of syntax directed translators, intermediate code, postfix notation, parse trees and syntax trees, three address code, quadruples, and triples, translation of assignment statements, Boolean expressions, control statements.

3. The contents of a symbol table, data structures for symbol tables, representing scope information. Implementation of a simple stack allocation scheme, implementation of block structured languages, storage allocation in block structured language. Error-lexical phase errors, syntactic phase errors, semantic errors.

4. The principle sources of optimization, loop optimization, the DAG representation of basic blocks, value number and algebraic laws, global data-flow analysis. Object programs, problems in code generation, a machine model, a simple code generator, register allocation and assignment, code generation from DAGs, peephole optimization.

Reference Books

1. Aho A.V. and Ullman J.D. Principles of Compiler Design, Addison Wesley.
2. Donovan, J, System Programming, TMH.
3. D.M. Dhamdhere: Compiler Construction-Principles and Practice McMillan India.
4. David Gries: Compiler Construction for Digital Computer.

Course Outcomes

1. Learn the fundamentals of language translator and compiler design.
2. Ability to design algorithms for Parsers (Top-down and Bottom-Up).
3. Classify and discuss intermediate code generation and optimization techniques to improve the performance of a program
4. Gain knowledge of techniques of symbol table organization, fundamentals of runtime environment and Code generation.

Course Code	**:**	**CSPE46**
Course Title	**:**	**Graph theory and Combinatorics**
Number of Credits	**:**	**4**
Prerequisites(Course Code)	**:**	**MAIR13**

Course Type	**:**	**PE**

Course Learning Objectives:

1. To introduce topics and techniques of discrete methods and combinatorial analysis.
2. To introduce a large variety of applications and, through some of them, the algorithmic approach to the solution of problems.

Course content

1. **Graph fundamentals:** Walks, paths, directed graphs, knots, cycles, neighborhood, clique, reachability, connectivity, spanning tree, knot detection algorithms.
   Graph coloring: Problem and model, greedy and random coloring algorithms, edge coloring, coloring trees, coloring planar and arbitrary graph.

2. **Graph traversal:** Tarry's traversal algorithm, search algorithms, BFS algorithms and applications, classical and Awerbuch's DFS algorithm, DFS with neighbor knowledge, spanning tree construction using flooding and DFS, GHS, Awerbuch, Korach-Kutten-Moran MST algorithms and applications, synchronizers.
   Matching: Unweighted and weighted matching algorithms, matching from edge coloring.
   Vertex cover: Unweighted and weighted vertex cover algorithms, pricing algorithm, bipartite matching-based algorithms.

3. **Routing:** Shortest path algorithms, Dijkstra, Bellman-Ford, Floyd-Warshall, Chandi-Misra algorithms, Johnson's algorithm, link state, distance vector, and interval routing algorithms, maximum flow, Ford-Fulkerson method, maximum bipartite matching, push-relabel algorithm, relabel-to-front algorithm.
   Maximal Independent Set (MIS): Rank-based MIS algorithms, Luby's MIS algorithm, MIS construction from vertex coloring.
   Domination algorithms: Greedy MDS algorithms, Guha-Khuller algorithm, connected dominating sets, MIS-based and pruning-based algorithms, weakly connected dominating sets.

4. **Network Algorithms:** Network analysis, vertex degrees, degree distribution, degree correlation, distance statistics, clustering coefficient, local and global view, centrality, Topology control.

Suggested Books:

1. Douglas B. West, Introduction to Graph Theory, PHI.
2. David Avis, Alain Hertz, Odile Marcotte (Eds.), Graph Theory and Combinatorial Optimization, Springer.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, PHI.
4. Sukumar Ghosh, Distributed Systems: An Algorithmic Approach, 2/EChapman & Hall/CRC.
5. KayhanErciyes, Distributed Graph Algorithms for Computer Networks, Springer.
   Yvonne-Anne Pignolet, Thomas Locher, Roger Wattenhofer, Principles of Distributed
6. Computing, Available online.
7. Herbert S. Wilf, Algorithms and Complexity, Available online.
8. Michel Raynal, Distributed Algorithms for Message-Passing Systems, Springer.
9. Nancy Lynch, Distributed Algorithms, Morgan Kaufmann.
10. Gerard Tel, Introduction to Distributed Algorithms, Cambridge Univ. Press

Course Outcome:

1. Able to apply knowledge of Discrete Mathematics to model computer networks.
2. Design distributed graph algorithms for complex networks.
3. Analyze the performance of various graph algorithms and evaluate the scalability of solutions for large and heterogeneous networks.
4. Adapt conventional graph algorithms to ad hoc networks and use knowledge of mathematics to sketch proof
5. Simulate the algorithms in wireless environment.

Open Elective – I

Course Code	**:**	**CSOE41**
Course Title	**:**	**Operating Systems**
Number of Credits	**:**	**3**
Prerequisites	**:**	
Course Type	**:**	**OE**

Course Learning Objectives

1. To understand the services and design of an operating system.
2. To understand the structure and organization of file system
3. To understand the process states and various concepts such as scheduling and synchronization related with it.
4. To understand different memory management approaches.
5. Students should be able to use system calls for managing processes, memory and file system.
6. students should understand the data structures and algorithms for implementation of OS.

Course Content

1. Computer system architecture and organization, Introduction and evolution of OS, Introduction to distributed OS, Real time systems and multimedia systems. OS structures: OS services, system calls and programs, OS design and implementation. Processes: Process concept, scheduling policies, algorithms, multilevel queuing, operations on process, Inter-process communication. Threads: multithreading models and threading issues. CPU scheduling: Criteria and algorithms, multiprocessor and thread scheduling.

2. Process synchronization: critical sections, classical two process and n-process solutions, hardware primitives for synchronization, semaphores, monitors, classical problems in synchronization (producer-consumer, readers-writer, dining philosophers, etc.). Deadlocks: modeling, resource allocation, characterization, prevention and avoidance, detection and recovery.

3. Memory management: Swapping, contiguous memory allocation, paging, multilevel paging, segmentation, demand paging, page replacement algorithms, allocation of frames, thrashing, working set model. Input/Output: I/O system and services, device controllers and device drivers, disks, scheduling algorithms and management.

4. File system interface: access methods, access control, directory structures, file organization, file sharing and protection. system performance, protection and security, access matrix, Security breaches, types of attacks, attack prevention methods, security policy and access control, OS design considerations for security, access, policy and access control, OS design considerations for security, access control lists and OS support, internet and general network security.

Reference Books:

1. A. Silberschatz, Peter B. Galvin and G. Gagne, "Operating System Concepts," (6th or newer edition), Wiley.
2. H. Brinch, "Operating System Principles," Prentice Hall of India.
3. Dhamdhere, "Systems programming & Operating systems," TataMcGraw Hill.
4. A. N. Habermann, "Introduction to Operating System Design," Galgotia publication, New Delhi.
5. A.S. Tanenbaum, "Modern Operating Systems," Prentice Hall of India.

Course outcomes

1. At the end of the course student will be able to
2. Understand functions, structures and history of operating systems
3. Able to know the design issues associated with operating systems

4. Master various process management concepts such as scheduling, synchronization, multithreading and deadlocks
5. Understand the various concepts associated with memory management such as virtual memory, demand paging, page replacements algorithms
6. Be familiar with various protection and security mechanisms

Open Electives – II and III

Course Code	**:**	**CSOE40**
Course Title	**:**	**Database Systems**
Number of Credits	**:**	**3**
Prerequisites	**:**	
Course Type	**:**	**OE**

Course Learning Objectives:

1. To understand fundamentals of data models and to conceptualize a database system for user requirement.
2. To study the fundamental of Database query language, like SQL and relational algebra.
3. To learn the concept of normalization in database design.
4. To learn fundamental concepts of transaction processing, concurrency control techniques and database recovery procedure.
5. Understand the professional, ethical and security issues and responsibilities in database design.

Course Content

1. **Introduction:** Overview of file systems and Database Systems , Software Architecture of a typical DBMS, Data Models, Relational Data Model, Network Data Model, Hierarchical Data Model, Schemas and Instances, Database Administration and Control.

2. **Relational Model and Query Language**: Overview of Relational Database, Key Integrity constraint, Relational Algebra, Relational Calculus: Domain Relational Calculus, Tuple Relational Calculus, SQL Fundamentals, Basic operators, Missing Information, Null Values, Additional Operations, Advanced SQL features, Embedded SQL, Dynamic SQL, Database Views.

3. **Database Design**: Overview of Normalization, Database Anomalies, Functional Dependencies, Candidate and Super Key, Non-loss Decomposition, Dependency Preservation, Normal forms: First, Second, Third Normal, Boyce Codd Normal Form, Multi-valued Functional Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form, Denormalization.

4. **Transaction Processing**: Overview of Database Transactions, Transaction states, Transaction Recovery, ACID Properties, Transaction Recovery: Two Phase Commit, Commit Points, Serializability, Concurrency control: Need for Concurrency, Locking Protocols: Binary lock, Two Phase Lock, Deadlock, Starvation, Transaction Timestamp.

Reference Books

1. A Silberschatz, H.F. Korth & S. Sudarshan: Data Base System Concepts, TMH, 1997.
2. A.K. Majumdar and Bhattacharyya: Database Management Systems, THM, 1996.
3. C.J. Date: An Introduction to Database systems 7th Ed. Addison Wesley, Indian Edition, 2000.
4. Elmasri & Navathe : Fundamentals of Database Systems/Oracle 9i Programming 5th Ed. Pearson, 2009

Course Outcomes

1. To understand fundamental concepts, principles and applications of database system.
2. To demonstrate database related programming languages and perform the basics of commercial relational systems.
3. To apply the concepts of normalization on database design.

4. To Design and Implement a small database project, considering the issues like concurrency control, recovery and security.
5. Comprehend contemporary issues related to industry and government related to database domain.

Course Code	**:**	**CSOE42**
Course Title	**:**	**Soft Computing**
Number of Credits	**:**	**3**
Prerequisites	**:**	
Course Type	**:**	**OE**

Course Learning Objectives:

1. Motivation and historical background of Soft Computing.
2. Application of Fuzzy logic.
3. Biologically inspired algorithm such as neural networks, genetic algorithms, ant colony optimization, and bee colony optimization.
4. Hybrid systems of neural network, genetic algorithms and fuzzy systems.

Course content

1. **Soft Computing and Artificial Intelligence:** Introduction of Soft Computing, Soft Computing vs. Hard Computing, Various Types of Soft Computing Techniques, Applications of Soft Computing, AI Search Algorithm, Predicate Calculus, Rules of Interference, Semantic Networks, Frames, Objects, Hybrid Models.

2. **Artificial Neural Networks and Paradigms:** Introduction to Neuron Model, Neural Network Architecture, Learning Rules, Perceptrons, Single Layer Perceptrons, Multilayer Perceptrons, Back propagation Networks, Kohnen'sself organizing networks, Hopfield network, Applications of NN.

3. **Fuzzy Logic:** Introduction, Fuzzy sets and Fuzzy reasoning, Basic functions on fuzzy sets, relations, rule based models and linguistic variables, fuzzy controls, Fuzzy decision making, applications of fuzzy logic.

4. **Genetic Algorithms and Swarm Optimizations:** Introduction, Genetic Algorithm, Fitness Computations, Cross Over, Mutation, Evolutionary Programming, Classifier Systems, Genetic Programming Parse Trees, Variants of GA, Applications, Ant Colony Optimization, Particle Swarm Optimization, Artificial Bee Colony Optimization.

Suggested Books :-

1. Simon S. Haykin, Neural Networks, Prentice Hall, 2nd edition.
2. B. Yegnanrayana , “Artificial Neural Networks” , PHI.
3. Jacek M. Zurada, Introduction to Artificial Neural Systems, Jaico Publishing House,1994
4. Zimmermann, “Fuzzy Set Theory and its Application”, 3rd Edition.
5. Jang J.S.R., Sun C.T. and Mizutani E, "Neuro-Fuzzy and Soft computing", Prentice Hall, 1998.
6. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", McGraw Hill, 1997.
7. D.E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley, N.Y, 1989.

Course Outcomes

1. Understand various biologically inspired algorithms.
2. Apply different intelligent algorithms in classification, prediction, optimization, pattern recognition applications.
3. Design hybrid system incorporating neural network, genetic algorithms, fuzzy systems.