B.TECH. COMPUTER SCIENCE AND ENGINEERING - 2022

Year	THIRD SEMESTER						FOURTH SEMESTER					
	Subject Code	**Subject Name**	**L**	**T**	**P**	**C**	**Subject Code**	**Subject Name**	L	T	P	C
II	MAT 2126	Engineering Mathematics - III	2	1	0	3	MAT 2226	Engineering Mathematics - IV	2	1	0	3
	CSE 2121	Computer Organization & Architecture	3	1	0	4	CSE 2221	Formal Languages and Automata Theory	2	1	0	3
	CSE 2122	Data Structures	3	1	0	4	CSE 2222	Design and Analysis of Algorithms	3	1	0	4
	CSE 2123	Digital System Design	3	1	0	4	CSE 2223	Embedded Systems	4	0	0	4
	CSE 2124	Object Oriented Programming	3	1	0	4	CSE 2224	Database Systems	3	1	0	4
	CSE 2141	Data Structures Lab	0	0	3	1	CSE 2241	Database Systems Lab	0	0	3	1
	CSE 2142	Digital System Design Lab	0	0	3	1	CSE 2242	Algorithms Lab	0	0	3	1
	CSE 2143	Object Oriented Programming Lab	0	0	3	1	CSE 2243	Embedded Systems Lab	0	0	3	1
			14	**5**	**9**	**22**			**14**	**4**	**9**	**21**
	Total Contact Hours (L + T + P)		**28**				**Total Contact Hours (L + T+ P)**		**27**			
	FIFTH SEMESTER						SIXTH SEMESTER					
III	HUM 3021	Essentials of Management	3	0	0	3	HUM 3022	Engineering Economics and Financial Management	3	0	0	3
	CSE 3121	Principles of Cryptography	2	1	0	3	CSE 3221	Parallel Computer Architecture and Programming	2	1	0	3
	CSE 3122	Software Engineering	2	1	0	3	CSE 3222	Compiler Design	2	1	0	3
	CSE 3123	Operating Systems	2	1	0	3	CSE ****	Program Elective – I/ (Minor Specialization)	3	0	0	3
	CSE 3124	Computer Networks	2	1	0	3	CSE ****	Program Elective – II /(Minor Specialization)	3	0	0	3
	IPE 4302	Open Elective -1 Creativity, Problem Solving and Innovation	3	0	0	3	*** ****	Open Elective- 2	3	0	0	3
	CSE 3141	Software Engineering Lab	0	0	3	1	CSE 3241	Parallel Programming Lab	0	0	3	1
	CSE 3142	Operating Systems Lab	0	0	3	1	CSE 3242	Compiler Design Lab	0	0	3	1
	CSE 3143	Computer Networks Lab	0	0	3	1	CSE 3243	Web Programming Lab	1	0	2	1
			14	**4**	**9**	**21**			**17**	**2**	**8**	**21**
	Total Contact Hours (L + T + P)		**27**				**Total Contact Hours (L + T + P)**		**27**			
	SEVENTH SEMESTER						EIGHTH SEMESTER					
IV	CSE ****	Program Elective – III / (Minor Specialization)	3	0	0	3	CSE 4291	Industrial Training				1
	CSE ****	Program Elective – IV/ (Minor Specialization)	3	0	0	3	CSE 4292	Project Work / Practice School				12
	CSE ****	Program Elective – V	3	0	0	3	CSE 4293	Project Work (B. Tech Honours) **				20
	CSE ****	Program Elective - VI	3	0	0	3	CSE ****	B Tech Honours (Theory 1)** (V Semester)				4
	CSE ****	Program Elective - VII	3	0	0	3	CSE ****	B Tech Honours (Theory 2)** (VI Semester)				4
	*** ****	Open Elective-3	3	0	0	3	CSE ****	B Tech Honours (Theory 3)** (VII Semester)				4
	CSE 4191	Mini Project (Minor Specialization) *				8						
			18	**0**	**0**	**18/26**						**13/33**
	Total Contact Hours (L + T + P)		**18**									

*Applicable to students who opted for minor specialization

**Applicable to eligible students who opted for and successfully completed the B Tech – Honours requirements

Minor Specializations

I. Computer Graphics & Visualization

CSE 4401: Digital Image Processing
CSE 4402: Computer Graphics
CSE 4403: Computer Vision
CSE 4404: Augmented and Virtual Reality

II. Computational Intelligence

CSE 4405: Artificial Intelligence
CSE 4406: Soft Computing Paradigms
CSE 4403: Computer Vision
CSE 4408: Machine Learning

III. Internet of Things

CSE 4409: Introduction to IoT
CSE 4410: IoT in Agriculture
CSE 4411: IoT for Healthcare
CSE 4412: Smart Cities

IV. Data Analytics

CSE 4413: Data Warehouse and Data Mining
CSE 4414: Natural Language Processing
CSE 4403: Computer Vision
CSE 4416: Big Data Analytics

V. Cyber Security

CSE 4417: Network Security
CSE 4418: Cyber forensics
CSE 4419: Artificial Intelligence in Cyber security
CSE 4420: Database and Application Security

Other Programme Electives

CSE 4441: Advanced Algorithms
CSE 4442: Android Application Development
CSE 4443: Animation Technologies
CSE 4444: Blockchain Technology
CSE 4445: Cloud Computing
CSE 4446: Cryptanalysis
CSE 4447: Deep Learning
CSE 4448: Distributed Systems
CSE 4449: Ethical Hacking and Cyber Security
CSE 4450: Game Programming
CSE 4451: Hardware Security CSE 4452: High Performance Computer Architecture
CSE 4453: Human Computer Interface
CSE 4454: Information Retrieval
CSE 4455: Information Security
CSE 4456: iOS Application Development
CSE 4457: Knowledge Representation and Ontology
CSE 4458: Machine Translation
CSE 4459: Multimedia Retrieval
CSE 4460: Multimedia Technologies
CSE 4461: Pattern Anomaly and detection
CSE 4462: Pervasive Computing
CSE 4463: Quantum Computing
CSE 4464: Social Network Analysis
CSE 4465: Software Architecture
CSE 4466: Software Defined Networks
CSE 4467: Software Testing and Analysis
CSE 4468: Storage Device and Technology
CSE 4469: UML and Design Patterns
CSE 4470: Wireless Networks

Open Electives

CSE 4311: Essentials of Industrial Computing
CSE 4312: Essentials of IT
CSE 4313: Linux Programming
CSE 4314: Principles of Database Systems
CSE 4315: Principles of Software Engineering
CSE 4316: Python Programming
CSE 4317: Web Programming
CSE 4318: iOS Application Development

Inter-Institute Elective (IIE)

IIE ****: Healthcare IT

B.TECH. COMPUTER SCIENCE AND ENGINEERING

Year	THIRD SEMESTER						FOURTH SEMESTER					
	Subject Code	**Subject Name**	**L**	**T**	**P**	**C**	**Subject Code**	**Subject Name**	L	T	P	C
II	MAT 2126	Engineering Mathematics - III	2	1	0	3	MAT 2226	Engineering Mathematics - IV	2	1	0	3
	CSE 2121	Computer Organization & Architecture	3	1	0	4	CSE 2221	Formal Languages and Automata Theory	2	1	0	3
	CSE 2122	Data Structures	3	1	0	4	CSE 2222	Design and Analysis of Algorithms	3	1	0	4
	CSE 2123	Digital System Design	3	1	0	4	CSE 2223	Embedded Systems	4	0	0	4
	CSE 2124	Object Oriented Programming	3	1	0	4	CSE 2224	Database Systems	3	1	0	4
	CSE 2141	Data Structures Lab	0	0	3	1	CSE 2241	Database Systems Lab	0	0	3	1
	CSE 2142	Digital System Design Lab	0	0	3	1	CSE 2242	Algorithms Lab	0	0	3	1
	CSE 2143	Object Oriented Programming Lab	0	0	3	1	CSE 2243	Embedded Systems Lab	0	0	3	1
			14	**5**	**9**	**22**			**14**	**4**	**9**	**21**
	Total Contact Hours (L + T + P)		**28**				**Total Contact Hours (L + T+ P)**		**27**			
	FIFTH SEMESTER						SIXTH SEMESTER					
III	HUM 3021	Essentials of Management	3	0	0	3	HUM 3022	Engineering Economics and Financial Management	3	0	0	3
	CSE 3121	Principles of Cryptography	2	1	0	3	CSE 3221	Parallel Computer Architecture and Programming	2	1	0	3
	CSE 3122	Software Engineering	2	1	0	3	CSE 3222	Compiler Design	2	1	0	3
	CSE 3123	Operating Systems	2	1	0	3	CSE ****	Program Elective – I/ (Minor Specialization)	3	0	0	3
	CSE 3124	Computer Networks	2	1	0	3	CSE ****	Program Elective – II /(Minor Specialization)	3	0	0	3
	IPE 4302	Open Elective -1 Creativity, Problem Solving and Innovation	3	0	0	3	*** ****	Open Elective- 2	3	0	0	3
	CSE 3141	Software Engineering Lab	0	0	3	1	CSE 3241	Parallel Programming Lab	0	0	3	1
	CSE 3142	Operating Systems Lab	0	0	3	1	CSE 3242	Compiler Design Lab	0	0	3	1
	CSE 3143	Computer Networks Lab	0	0	3	1	CSE 3243	Web Programming Lab	1	0	2	1
			14	**4**	**9**	**21**			**17**	**2**	**8**	**21**
	Total Contact Hours (L + T + P)		**27**				**Total Contact Hours (L + T + P)**		**27**			
	SEVENTH SEMESTER						EIGHTH SEMESTER					
IV	CSE ****	Program Elective – III / (Minor Specialization)	3	0	0	3	CSE 4291	Industrial Training				1
	CSE ****	Program Elective – IV/ (Minor Specialization)	3	0	0	3	CSE 4292	Project Work / Practice School				12
	CSE ****	Program Elective – V	3	0	0	3	CSE 4293	Project Work (B. Tech Honours) **				20
	CSE ****	Program Elective - VI	3	0	0	3	CSE ****	B Tech Honours (Theory 1)** (V Semester)				4
	CSE ****	Program Elective - VII	3	0	0	3	CSE ****	B Tech Honours (Theory 2)** (VI Semester)				4
	*** ****	Open Elective-3	3	0	0	3	CSE ****	B Tech Honours (Theory 3)** (VII Semester)				4
	CSE 4191	Mini Project (Minor Specialization) *				8						
			18	**0**	**0**	**18/26**						**13/33**
	Total Contact Hours (L + T + P)		**18**									

*Applicable to students who opted for minor specialization

**Applicable to eligible students who opted for and successfully completed the B Tech – Honours requirements

Minor Specializations

I. Computer Graphics & Visualization

CSE 4401: Digital Image Processing
CSE 4402: Computer Graphics
CSE 4403: Computer Vision
CSE 4404: Augmented and Virtual Reality

II. Computational Intelligence

CSE 4405: Artificial Intelligence
CSE 4406: Soft Computing Paradigms
CSE 4403: Computer Vision
CSE 4408: Machine Learning

III. Internet of Things

CSE 4409: Introduction to IoT
CSE 4410: IoT in Agriculture
CSE 4411: IoT for Healthcare
CSE 4412: Smart Cities

IV. Data Analytics

CSE 4413: Data Warehouse and Data Mining
CSE 4414: Natural Language Processing
CSE 4403: Computer Vision
CSE 4416: Big Data Analytics

V. Cyber Security

CSE 4417: Network Security
CSE 4418: Cyber forensics
CSE 4419: Artificial Intelligence in Cyber security
CSE 4420: Database and Application Security

Other Programme Electives

CSE 4441: Advanced Algorithms
CSE 4442: Android Application Development
CSE 4443: Animation Technologies
CSE 4444: Blockchain Technology
CSE 4445: Cloud Computing
CSE 4446: Cryptanalysis
CSE 4447: Deep Learning
CSE 4448: Distributed Systems
CSE 4449: Ethical Hacking and Cyber Security
CSE 4450: Game Programming
CSE 4451: Hardware Security CSE 4452: High Performance Computer Architecture
CSE 4453: Human Computer Interface
CSE 4454: Information Retrieval
CSE 4455: Information Security
CSE 4456: iOS Application Development
CSE 4457: Knowledge Representation and Ontology
CSE 4458: Machine Translation
CSE 4459: Multimedia Retrieval
CSE 4460: Multimedia Technologies
CSE 4461: Pattern Anomaly and detection
CSE 4462: Pervasive Computing
CSE 4463: Quantum Computing
CSE 4464: Social Network Analysis
CSE 4465: Software Architecture
CSE 4466: Software Defined Networks
CSE 4467: Software Testing and Analysis
CSE 4468: Storage Device and Technology
CSE 4469: UML and Design Patterns
CSE 4470: Wireless Networks

Open Electives

CSE 4311: Essentials of Industrial Computing
CSE 4312: Essentials of IT
CSE 4313: Linux Programming
CSE 4314: Principles of Database Systems
CSE 4315: Principles of Software Engineering
CSE 4316: Python Programming
CSE 4317: Web Programming
CSE 4318: iOS Application Development

Inter-Institute Elective (IIE)

IIE ****: Healthcare IT

THIRD SEMESTER

MAT 2126: ENGINEERING MATHEMATICS III [2103]

(Common to CSE / ICT/CC)

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

References:

1. C. L. Liu : *Elements of Discrete Mathematics*, 2nd edition, 2007, Mc Graw Hill, New Delhi.
2. J. P. Trembaly and R. Manohar: *Discrete Mathematics Structures with application to computer science*, 2012,Tata Mc Graw Hill.
3. E. S. Page and L. B. Wilson : *An introduction to computational combinatroics*, edn., 1979, Cambridge Univ. Press.
4. Narasingh Deo : *Graph theory with Applications to computer science*, PHI, 2012.
5. F. Harary, *Graph Theory*, Narosa Publishing House, New Delhi, Second edition, 1990.

CSE 2121: COMPUTER ORGANIZATION AND ARCHITECTURE
[3 1 0 4]

Course outcomes:

After studying this course, students will be able to:

1. Interpret the functionalities of the various units of computers and design programs using the instruction set architecture
2. Apply the hardware implementation of addition, subtraction, multiplication and division to perform arithmetic operations.
3. Design the control unit for basic algorithms.
4. Apply the concepts of memory system for efficient memory design.

5. Outline the I/O handling techniques and realize the improvement in the performance using concepts of pipelining and parallel processing.

Number and Character Representation, Arithmetic Operations. Memory location, addresses, and operations, Instructions and Instruction Sequencing, Addressing modes, RISC and CISC, Hardware for addition and subtraction, Multiplication, Booth's algorithm, Division, IEEE standard floating-point representation, Floating point arithmetic. Bus organization, Design methods- hardwired and micro-programmed approach for booths multiplier. Internal organization of memory chips, Memory Hierarchy, Cache memories, Virtual memories, Magnetic hard disk, I/O Device Interface, Program-Controlled I/O, Interrupts, Direct Memory Access, Direct Cache Access, I/O Channels and Processors, Interconnection Standards. Pipelining, Hardware Multithreading, Vector Processing, Graphics Processing Units, Shared-Memory Multiprocessors, Cache Coherence, Directory-Based Cache Coherence

References:

1. Carl Hamacher, ZvonkoVranesic and SafwatZaky, Computer Organization and Embedded Systems, (6e), McGraw Hill Publication, 2012
2. William Stallings, Computer Organization and Architecture, 10th Edition, Pearson, 2017
3. Mohammed Rafiquzzaman and Rajan Chandra, Modern Computer Architecture, Galgotia Publications Pvt. Ltd., 2010
4. D.A. Patterson and J.L.Hennessy, Computer Organization and Design-The Hardware/Software Interface, (5e), Morgan Kaufmann, 2014
5. J.P.Hayes, Computer Architecture and Organization, McGraw Hill Publication, 1998

CSE 2122: DATA STRUCTURES [3 1 0 4]

Course outcomes:

After studying this course, students will be able to:

1. Illustrate the usage of user defined data types and dynamic memory management functions.
2. Design applications using queues and stacks.
3. Develop applications using linked lists.
4. Apply concepts of trees for hierarchical data representation.
5. Outline graph representations and graph traversals

Recursion, Introduction - Pointers and Pointer Application, Accessing variables through pointers, pointers to pointers, pointer arithmetic and arrays, pointers and functions, Stacks, queues, evaluation of expressions, Linked lists representations- Singly, doubly, header node, circular along with the applications, Trees-Binary trees, representation, recursive/ non recursive inorder, preorder and post order tree traversal, level order traversal, Binary search tree, creation, insertion deletion operations on binary search tree, Additional Binary Tree Operations, Threaded Binary Tree and applications, Graphs – Representation and Traversals.

References:

1. Behrouz A. Forouzan, Richard F. Gilberg, A Structured Programming Approach Using C,(3e), Cengage Learning India Pvt. Ltd, India, 2007.
2. Ellis Horowitz, Sartaj Sahani, Susan Anderson and Freed, Fundamentals of Data Structures in C, (2e), Silicon Press, 2007.
3. Richard F. Gilberg, Behrouz A. Forouzan, Data structures, A Pseudocode Approach with C, (2e), Cengage Learning India Pvt. Ltd, India, 2009.
4. Tenenbaum Aaron M., Langsam Yedidyah, Augenstein Moshe J., Data structures using C, Pearson Prentice Hall of India Ltd., 2007.
5. Debasis Samanta, Classic Data Structures, (2e), PHI Learning Pvt. Ltd., India, 2010.

CSE 2123: DIGITAL SYSTEM DESIGN [3 1 0 4]

Course outcomes:

After studying this course, students will be able to:

1. Identify and explain the fundamental concepts of basic gates, implementation technology and apply K-map to simplify logical expressions.
2. Construct and analyze arithmetic circuits and combinational circuits using multiplexers, encoders, and decoders
3. Discuss about the types of flip-flops and create algorithmic state machine charts
4. Design synchronous and asynchronous sequential circuits for different applications
5. Build systems like simple processor, bit counting circuit

Brief overview of Logic gates, Truth Tables, Sum-of-Products and Product-of-Sums forms, K-Map Simplification, Incompletely Specified Functions, Fan-in, Factoring, Functional decomposition, Multilevel NAND and NOR Circuits, Addition of unsigned

and signed numbers, BCD Adder, Fast adder, Array multiplier, Multiplexer, Decoder, Encoder, Code converter, Arithmetic comparison circuits, Flip-Flops, Design of Synchronous Sequential Circuits-State reduction, State assignment, Ripple Counters, Registers, Shift Registers, Ring and Johnson Counters, Verilog for combinational and sequential circuits, Algorithmic State Machine Charts, Transistor Switches, NMOS, CMOS Logic Gates, Programmable Logic Devices, Transmission Gates, Tri-state drivers, Practical aspects, Simple processor and Bit count circuit design.

References

1. Stephen Brown and Zvonko Vranesic, *Fundamentals of Digital Logic with Verilog Design* (3e), Tata McGraw Hill, 2014.
2. Morris Mano M. and Michael D. Ciletti., Digital Design: With an introduction to the Verilog HDL (5e), PHI Learning 2007.
3. Morris Mano M., Digital Design, (2e), PHI Learning, 2000
4. Donald D. *Givone, Digital Principles and Design*, Tata McGraw Hill, 2003
5. John F. Wakerly, *Digital design - Principles and practice,* (4e), Pearson Education, 2013

CSE 2124: OBJECT ORIENTED PROGRAMMING [3 1 0 4]

Course outcomes:

After studying this course, students will be able to:

1. Utilize knowledge of basic Object oriented programming features along with CERT JAVA coding standard to solve real world problems.
2. Apply inheritance and packages to ensure code reusability.
3. Create reusable components using Generics.
4. Develop robust and efficient programs using exception handling and multithreading
5. Design GUI based applications using JavaFX

Introduction to OOP, Java Programming Fundamentals, Data types & Operators, Control structures, strings, Introducing Classes, Objects and Methods, Inheritance: Inheritance basics, Constructors, Interfaces: Fundamentals, creating and implementing an interface, Packages: Fundamentals, packages and member access, Exception handling: Exception hierarchy and fundamentals, try block, multiple catch clauses, throw and throws, finally, user defined exceptions, Multithreaded Programming: Multithreading fundamentals, creating threads, thread priorities, synchronization, thread communication, Generics:

Generic fundamentals, Generic class, bounded types, wildcards, Generic methods, Generic restrictions, GUI Programming with Javafx: Introducing Javafx: Basic concepts, Application Skeleton, Using buttons and events, Exploring Javafx Controls, CERT Java Coding Standard: Rules and Recommendations.

References:

1. Herbert Schildt and Dale Skrien, *Java Fundamentals – A Comprehensive Introduction*, (1e), McGrawHill, 2015
2. Herbert Schildt, *Java The Complete Reference*, (10e), Tata McGrawHill, 2017
3. Fred Long, Dhruv Mohindra, *Ebook:CERT Oracle Secure Coding Standard for Java*, Addison Wesley, 2013
4. Fred Long, Dhruv Mohindra, *Ebook:Java Coding Guidelines: 75 Recommendations for Reliable and Secure Programs*, Addison Wesley, 2014
5. Herbert Schildt, *Java A beginner's Guide*, (6e), 2014

CSE 2141: DATA STRUCTURES LAB [0 0 3 1]

Course outcomes:

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

1. Develop programs using user defined data types and dynamic memory management functions.
2. Design programs using stacks and queues.
3. Solve computing problems using linked lists and trees.

Reviewing the concepts of pointers, structures and recursion, Studying the operation of stacks and queues and the associated application programs, Creating dynamic allocation of memory for linked list and applying it to examples using singly, doubly and circular linked list and their applications, Creation of binary trees and the application associated with the trees.

References:

1. Behrouz A. Forouzan, Richard F. Gilberg, *A Structured Programming Approach Using C*, (3e), Cengage Learning India Pvt. Ltd, India, 2007
2. Ellis Horowitz, Sartaj Sahni, Susan Anderson and Freed, *Fundamentals of Data Structures in C*, (2e), Silicon Press, 2007

3. Richard F. Gilberg, Behrouz A. Forouzan, *Data structures, A Pseudocode Approach with C*, (2e), Cengage Learning India Pvt. Ltd, India, 2009
4. Tenenbaum Aaron M., Langsam Yedidyah, Augenstein Moshe J., *Data structures using C*, Pearson Prentice Hall of India Ltd., 2007
5. Debasis Samanta, *Classic Data Structures*, (2e), PHI Learning Pvt. Ltd., India, 2010

CSE 2142: DIGITAL SYSTEM DESIGN LAB [0 0 3 1]

Course outcomes:

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

1. Demonstrate the simulation of logical and arithmetic circuits in Verilog.
2. Develop the Verilog code for multiplexers, encoders, decoders and utilize them in the hierarchical code to build various applications.
3. Design and simulate sequential circuits and simple processors using Verilog.

Simulation of Logic Circuits Using Verilog: Verification of Logic Gates and logic expressions, Simplification of Expressions using Kmap: SOP and POS Forms, Multilevel NAND, NOR Circuits, Arithmetic Circuits: Half Adder, Full Adder, Multi-Bit Adder/Subtractor, BCD Adder, Multiplexers, Decoders and Encoders, Code Converters and Comparator, Latches and Flip-Flops: D, JK, and T Flip-Flops, Registers: Shift Register, Design of sequential circuits, Ring Counter, Johnson Counter, Binary Counters, Simple processor design

References:

1. Stephen Brown and Zvonko Vranesic, *Fundamentals of Digital Logic with Verilog Design*, (3e), Tata McGraw Hill, 2014
2. Morris Mano M., *Digital Design*, (2e), PHI Learning 2000

CSE 2143: OBJECT ORIENTED PROGRAMMING LAB [0 0 3 1]

Course outcomes:

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

1. Apply knowledge of basic object oriented programming features to solve real world problems.
2. Create reusable code using inheritance, packages and generics.
3. Design and develop Java applications using exception handling, multithreading and graphical user interface.

Simple Java programs using control structures and Arrays, Programs using Classes, objects, methods, Programs on Constructors and static members, Programs using Inheritance, Packages, Interfaces and Generics, Programs using Exceptions and Multithreading, GUI based programs using Javafx.

References:

1. Herbert Schildt and Dale Skrien, *Java Fundamentals – A Comprehensive Introduction,* (1e), McGrawHill, 2015
2. Herbert Schildt, *The Complete Reference JAVA 2,* (10e), Tata McGrawHill, 2017
3. Dietel and Dietel, *Java How to Program,* (9e), Prentice Hall India, 2012

FOURTH SEMESTER

MAT 2226: ENGINEERING MATHEMATICS IV [2103]

(Common to CSE / ICT/CC)

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

References:

1. P.L.Meyer : *Introduction to probability and Statistical Applications*,2nd edition, 1980, Oxford and IBH publishing, Delhi.
2. Miller, Freund and Johnson, *Probability and Statistics for Engineers*, 8th Edn, PHI, 2011.
3. Hogg and craig , *Introduction to mathematical statistics*, 6th Edn,2012, Pearson education, New Delhi.
4. Ross Sheldon M, *Introduction to Probability and Statistics for Engineers and Scientists*, Elseveir, 2010.

CSE 2221: FORMAL LANGUAGES AND AUTOMATA THEORY [2 1 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Demonstrate fundamental concept of grammars, languages and automata.
2. Design grammar and automata for regular languages.
3. Construct grammar and design automata for context free languages.
4. Build Turing machine for formal language classes.
5. Outline the theory of computation and computational models.

Three Basic concepts, Some Applications, DFA, NFA, Equivalence of DFA and NFA, State Reduction, Regular Expressions, Connection between regular expressions and regular languages, regular grammars, Closure properties of Regular Languages, Identifying Non-regular languages, Context-Free grammars, Parsing and Ambiguity, Methods for transforming Grammars, Two Important Normal Forms, NPDA, Push Down Automata and Context-Free Languages, DPDA, Pumping Lemma for Context Free Languages and Linear Languages, Closure properties and Decision Algorithms for Context-Free Languages, The Standard Turing Machine, Nondeterministic Turing Machines, A Universal Turing Machine, Recursive and Recursively Enumerable Languages, Unrestricted grammars, Context-Sensitive Grammars and Languages, Chomsky Hierarchy.

References:

1. Peter Linz, *An Introduction to Formal Languages and Automata, (6e),* Jones & Bartlett Learning, 2019.
2. J E Hopcroft, Rajeev Motwani & Jeffrey D Ullman, *Introduction to Automata Theory, Languages and Computation*, (3e), Pearson Education, 2006.
3. John C Martin, *Introduction to Languages and the Theory of Computation*, (3e), McGraw Hill, India, 2007.
4. Rajendra Kumar, *Theory of Automata, languages and computation*, Tata McGraw-Hill Education, 2010
5. K.L.P. Mishra, N.Chandrashekharan, *Theory of Computer Science, (3e),* PHI publications, 2007.

CSE 2222: DESIGN AND ANALYSIS OF ALGORITHMS [3 1 0 4]

Course outcomes:

After studying this course, students will be able to:

1. Apply techniques for designing algorithms to solve problems
6. Discuss different strategies of algorithm design techniques.
7. Demonstrate the complexity of algorithm using asymptotic notations
8. Analyse the various aspects which contribute to algorithm efficiency
9. Classify the complexity of algorithms into different efficiency classes

Fundamentals of Algorithms, Important Problem Types, Analysis of algorithm efficiency. Analysis Framework: Asymptotic Notations and Basic Efficiency Classes, Mathematical

Analysis of Non-recursive and Recursive Algorithms. Brute force Techniques, Divide and Conquer, Decrease and Conquer: Insertion Sort, Depth First Search, Breadth First Search, Topological Sorting. Transform and Conquer: Presorting, BST, Heapsort. Space and Time tradeoffs: Input Enhancement in String Matching. Dynamic Programming: Warshall's and Floyd's Algorithms, The Knapsack Problem. Greedy Techniques: Prim's, Kruskal's and Dijkstra's Algorithm, Huffman Trees. Coping with limitations of algorithmic power, P, NP, and NP-complete Problems, Backtracking: n–Queens problem, Hamiltonian Circuit Problem, Subset–Sum Problem. Branch and Bound: Assignment Problem, Knapsack Problem, TSP.

References:

1. Anany Levitin, *Introduction to the Design and Analysis of Algorithms,* (3e), Pearson Education, 2011
2. Ellis Horowitz and Sartaj Sahni, *Computer Algorithms/C++,* (2e), University Press, 2007.
3. Thomas H. Cormen, Charles E. Leiserson, Ronal L, Rivest, Clifford Stein, *Introduction to Algorithms,* (2e), PHI, 2006

CSE 2223: EMBEDDED SYSTEMS [4 0 0 4]

Course outcomes:

After studying this course, students will be able to:

1. Explain the architecture and addressing modes of ARM Cortex-M microcontroller.
2. Develop assembly language programs using ARM instruction set.
3. Compare and contrast different types of stack in ARM
4. Create the embedded C code for interfacing I/O devices such as LEDs, seven segment, LCD, keyboard, DC motor, stepper motor, UART, ADC and DAC.
5. Demonstrate timer/counter and interrupt programming.

Introduction to Embedded Systems, Microprocessors and Microcontrollers, An overview of ARM-Cortex- M Architecture, ARM memory map, CISC versus RISC, ARM addressing modes, Assembly language programming in ARM, Data transfer instructions, Arithmetic and logical instructions, Shift and rotate instructions, Branch and conditional branch instructions, subroutine call and return, Recursive functions, Conditional execution, Stack, Input/output (I/O) programming, Timer/counter programming, I/O interfacing - LED, LCD, Keyboard, Stepper motor, ADC, and DAC, PWM, UART, Nested Vectored Interrupt Controller (NVIC), External hardware interrupts, IO interrupts, timer/counter interrupts.

References:

1. Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi, Janice Mazidi, *ARM Assembly Language Programming & Architecture* (2e), MicroDigitalEd, 2016.
2. Jonathan W. Valvano., *Embedded systems: Introduction to Arm(r) Cortex-M Microcontrollers (5e),* Createspace Independent publishing platform, June 2014.
3. Jonathan W. Valvano., *Embedded systems: real-time interfacing to ARM Cortex-M microcontrollers (4e),* Createspace Independent Publishing Platform, 2014.
4. UM10360, LPC 176x/5x User Manual, NXP Semiconductors, Rev. 3.1, 2014.
5. Toulson and Tim Wilmshurst., *Fast and Effective Embedded System Design applying the ARM mbed*, Elsevier, 2017.
6. Joseph V., *A definitive Guide to ARM Cortex-M3 and Cortex-M4 processors (3e),* Elsevier, 2014.

CSE 2224: DATABASE SYSTEMS [3 1 0 4]

Course outcomes:

After studying this course, students will be able to:

1. Interpret the basic concepts of database and relational model
2. Apply structured query language for data retrieval
3. Design databases using E-R models
4. Analyze and apply the normalization technique to decompose given relational schema into effective schema
5. Explain file organization methods and apply different transaction management techniques

Database-System Applications, Database Languages, Relational Databases, Database Design, Data Storage and Querying, Database Architecture, NoSQl, Data Sharding, Database Schemas, Keys, Relational Query Languages, Relational Operations, SQL Data Definition, SQL Data Types and Schemas, Integrity Constraints, Set Operations, Aggregate Functions, Overview of SQL Query Language, Basic Structure of SQL Queries, Join Expressions, Overview of the Design Process, The Entity-Relationship Model, Extended E-R Features, Reduction to Relational Schemas, Features of Good Relational Design, Atomic Domains and Normalization, File concepts, Indices Concept, Static Hashing, Dynamic Hashing, Comparison of Ordered Indexing and Hashing, Bitmap Indices, Transaction Concept, Failure Classification, Storage, Recovery and Atomicity, Recovery algorithm Lock-based protocols, deadlock handling, Timestamp based protocols, Validation based protocols. .

References:

1. Silberschatz, Korth, Sudarshan, Database System Concepts, (6e), McGrawHill, New York, 2011.
2. Pramod J Sadalage, Martin Fowler, NoSQL Distilled, Addison-Wesley, 2013
3. Ramez Elmasri and Shamkant Navathe, Durvasula V L N Somayajulu, Shyam K Gupta, Fundamentals of Database Systems, (6e), Pearson Education, United States of America, 2011

CSE 2241 DATABASE SYSTEMS LAB [0 0 3 1]

Course Outcomes :

At the end of this course, students will be able to

1. Utilize SQL for design and manipulation of database tables.
2. Build stored procedures, cursors and exception handlers.
3. Create triggers, functions and packages.

Data Definition Language, Data manipulation language, Basic database query operations, Integrity Constraints in SQL, Nested subqueries, Join Operations, Views, PL/SQL Basics, Exception Handling, Cursors, Stored procedures, Functions, Packages, Trigger, and project on design and development of application based on database concepts.

References

1. Silberschatz, Korth, Sudarshan, Database System Concepts, (6e) McGrawHill, 2011.
2. Ivan Bayross, SQL, PL/SQL, (3e) , BPB Publications
3. G, Reese, Database Programming with JDBC and Java, (2e), O'REILLY, 2000.

CSE 2242 : ALGORITHMS LAB [0 0 3 1]

Course Outcomes:

At the end of this course, students will be able to

1. Demonstrate the fundamentals of algorithmic problem solving.
2. Apply various algorithm design paradigms.
3. Analyze and categorize the algorithms into different efficiency classes.

Implement a doubly linked list & BST, GCD Techniques, Bubble sort, Selection sort, Linear search, String Matching, sorting algorithms, DFS, BFS, Topological sorting, AVL tree, 2-3 tree, Horspool algorithm, Open hash table, Floyd's algorithm, Warshall's algorithm, Greedy Techniques, Dijkstra's algorithm, Backtracking.

References:

1. Anany Levitin, *Introduction to the Design and Analysis of Algorithms,* (3e), Pearson Education, India, 2011.
2. Ellis Horowitz and Sartaj Sahni, *Computer Algorithms/C++,* (2e), University Press, 2007
3. Thomas H. Cormen, Charles E. Leiserson, Ronal L, Rivest, Clifford Stein, *Introduction to Algorithms,* (2e*),* PHI, 2006

CSE 2243: EMBEDDED SYSTEMS LAB [0 0 3 1]

Course outcomes:

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

1. Experiment with Keil μVision simulator for ARM cortex-M microcontroller assembly language programming using data transfer and arithmetic instructions.
2. Develop assembly language program in Keil μVision simulator using logical, branching and looping instructions for ARM cortex-M microcontroller.
3. Create embedded C program for ARM cortex-M microcontroller by interfacing various modules to ARM kit.

Basics of assembly language programming, data transfer, arithmetic operations, logical instructions, branch instructions, code conversion, packing and unpacking of BCD and ASCII data, sorting, searching, recursion, I/O interfacing of LEDs, LCD, keyboard, 7 segment display, stepper motor, DAC and ADC, PWM.

References:

1. Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi, Janice Mazidi, *ARM Assembly Language Programming & Architecture* (2e), MicroDigitalEd, 2016.
2. Jonathan W. Valvano., *Embedded systems: Introduction to Arm(r) Cortex-M Microcontrollers (5e),* Createspace Independent publishing platform, June 2014.
3. Jonathan W. Valvano., *Embedded systems: real-time interfacing to ARM Cortex-M microcontrollers (4e),* Createspace Independent Publishing Platform, 2014.
4. UM10360, LPC 176x/5x User Manual, NXP Semiconductors, Rev. 3.1, 2014.
5. Toulson and Tim Wilmshurst., *Fast and Effective Embedded System Design applying the ARM mbed*, Elsevier, 2017.
6. Joseph V., *A definitive Guide to ARM Cortex-M3 and Cortex-M4 processors (3e),* Elsevier, 2014.

FIFTH SEMESTER

HUM 3021: ESSENTIALS OF MANAGEMENT [3 0 0 3]

Definition of management and systems approach, Nature & scope. The Functions of managers, Principles of Management. Planning: Types of plans, steps in planning, Process of MBO, how to set objectives, strategies, policies and planning premises, Strategic planning process and tools. Nature and purpose of organizing, Span of management, factors determining the span, Basic departmentation, Line and staff concepts, Functional authority, Art of delegation, Decentralization of authority. HR theories of planning, Recruitment, Development and training. Theories of motivation, Special motivational techniques. Leadership – leadership behavior & styles, Managerial grid. Basic Control Process, Critical Control Points & Standards, Budgets, Non-budgetary control devices. Profit and Loss control, Control through ROI, Direct, Preventive control. PROFESSIONAL ETHICS - Senses of Engineering Ethics, Variety of moral issues, Types of inquiry, Moral dilemmas, Moral Autonomy, Kohlberg's theory, Gilligan's theory, Consensus and Controversy, Models of professional roles, Theories about right action, Self-interest, Customs and Religion, Uses of Ethical Theories. GLOBAL ISSUES - Managerial practices in Japan and USA & application of Theory Z. The nature and purpose of international business & multinational corporations, unified global theory of management, Entrepreneurship and writing business plans. Multinational Corporations, Environmental Ethics, Computer Ethics, Weapons Development, Engineers as Managers, Consulting Engineers, Engineers as Expert Witnesses and Advisers, Moral Leadership, Code of Conduct, Corporate Social Responsibility.

References:

1. Harold Koontz & Heinz Weihrich (2012), "Essentials Of Management", Mc Graw Hill, New Delhi.
2. Peter Drucker (2004), "The Practice Of Management", Harper And Row, New York.
3. Vasant Desai (2007), "Dynamics Of Entrepreneurial Development & Management", Himalaya Publishing House.
4. Poornima M Charantimath (2006), "Entrepreneurship Development", Pearson Education.
5. Mike W. Martin And Ronald Schinzinger (2003), "Ethics In Engineering", Tata Mcgraw Hill, New Delhi.
6. Govindarajan M, Natarajan S, Senthil Kumar V S (2004), "Engineering Ethics", Prentice Hall of India, New Delhi.

CSE 3121: PRINCIPLES OF CRYPTOGRAPHY [2 1 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Illustrate the cryptographic attacks, services, and classical symmetric ciphers.
2. Analyze the various block ciphers and modes of operation.
3. Utilize the concepts of number theory and pseudo random number generation.
4. Explain the public key cryptosystems.
5. Apply various data integrity algorithms.

Security Goals, Attacks, Services, Mechanisms, Symmetric Cipher Model, Block Ciphers and DES, Strength of DES, Block Cipher Design Principles. AES, Equivalent Inverse Cipher. Block Cipher Operation- Multiple Encryption and Triple DES, Electronic Codebook, Cipher Block Chaining Mode, Cipher Feedback Mode, Output Feedback Mode, Counter Mode, XTS-AES Mode for Block-Oriented Storage Devices, Format-Preserving Encryption. Euclidean Algorithm, Modular Arithmetic, Prime Numbers, Fermat and Euler theorems, Testing for Primality, Chinese Remainder theorem, Discrete Logarithms. Pseudorandom Number Generation, Stream Ciphers, RC4. Public Key Cryptography and RSA. D-H Key Exchange, ElGamal System. Cryptographic Hash Functions. Message Authentication Codes, Security of MACs, HMAC.

References:

1. William Stallings, *Cryptography and Network Security: Principles and Practice*, (7e), Prentice Hall, 2017.
2. Behrouz A. Forouzan and Debdeep Mukhopadhyay, *Cryptography and Network Security*, (2e), McGraw Hill, 2008
3. Atul Kahate, *Cryptography and Network Security*, Tata McGraw-Hill Publishing, 2008
4. Bruce Schneier, *Applied Cryptography-Protocols, Algorithms, and source code in C*, (2e), John Wiley & Sons, Inc., 2013

CSE 3122: SOFTWARE ENGINEERING [2 1 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Illustrate the basic concepts and software life cycle models
2. Analyze the requirements of the software project
3. Design the software project based on object-oriented concepts
4. Extend the software design using UML Modeling techniques
5. Apply the programming standards and evaluate the software using testing techniques

Evolution from an art form to an engineering discipline, Software development Projects, Exploratory style of software development, Emergence of software Engineering, Notable changes in software development practices. Computer Systems Engineering. A few basic concepts, Waterfall model and its extensions, Rapid Application Development, Agile development models, Spiral Model, A Comparison of different Life Cycle models, Case Studies. Requirement Gathering and Analysis, Software Requirement Specifications, Case Studies Formal Specification Techniques, Case Studies. Overview of the design Process, How to characterize a good software design? Cohesion and coupling, Layered arrangement of modules, Approaches to software design. Overview of SA/SD methodology, Structured analysis, Developing the DFD Model of a system, Case Studies Structured Design, Case Studies Detailed design, Design review, Case Studies. Basic object-orientation concepts, UML, UML diagrams, Use case model, Class diagrams, Interaction diagrams, Activity Diagram, State chart diagram, Case Studies Postscript, Design Patterns, An Object-Oriented Analysis and Design (OOAD) Methodology, Case Studies. Coding, Code review, Software Documentation, Testing, Unit Testing, Black-Box testing, White-Box Testing, Debugging, Program Analysis tools, Regression testing, Security testing, Robustness testing, Fuzzy testing, Integration testing, Testing OOP, System testing, Some general issues associated with testing.

References:

1. Rajib Mall, Fundamentals of Software Engineering (5e), PHI Learning, 2019.
2. Hans Van Vliet, *Software Engineering: Principles and Practice* (3e), Wiley India, 2012.
3. Roger S. Pressman, *Software Engineering - A Practitioner's Approach* (7e), McGrawHill International Edition, 2010.
4. Bernd Bruegge, Allen H. Dutoit, *Object-Oriented Software Engineering using UML Patterns and Java* (2e) , Pearson Publication, 2011.
5. Ian Sommerville, *Software Engineering* (9e), Addison-Wesley, 2011.
6. Nooper Davis, *Secure Software Development Life Cycle Processes*, Software Engineering Institute, Carnegie Mellon University, 2013.
7. Julie Cohen, Dan Plakosh, Kristi Keeler, *Robustness Testing of Software-Intensive Systems: Explanation and Guide*, Carnegie Mellon University, 2005.

CSE 3123: OPERATING SYSTEMS [2 1 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the concepts of threads and process.

2. Interpret the different types of process synchronization techniques
3. Analyze different CPU scheduling and deadlock algorithms
4. Examine the main memory and the virtual memory concepts
5. Summarize the file systems and protection related to a general operating system

What Operating Systems Do, Operating System Structure, Operating System Operations, Process Management, Memory Management, and Storage Management. Operating System Services, User and Operating System Interface, System Calls, Types of System Calls, System Programs, Operating System Structure, Virtual Machines, System Boot. Overview, Process Scheduling, Operations on Processes, Interprocess Communication. Overview, Multithreaded Models, Thread Libraries. Background, The Critical Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores. Basic Concepts, Scheduling Criteria, Scheduling, Thread Scheduling, Linux scheduling. System Model, Deadlock, Characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock Avoidance. Logical Versus Physical Address Space, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of Page Table. Background, Demand Paging, Copy-On-Write, Page Replacement, Allocation of Frames, Thrashing. Disk Structure, Disk Scheduling, Swap-Space Management. File Concept, Access Methods, Directory and Disk Structure, File System Mounting. Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix Implementation of Access Matrix.

References:

1. A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, (9e), Wiley and Sons (Asia) Pte Ltd, 2016
2. D.M. Dhamdhere, *Operating Systems A Concept based Approach,(2e),* Tata McGraw-Hill 2006
3. Deitel & Deitel , *Operating systems, (3e),* Pearson Education, India. 2018.
4. Neil Matthew, Richard Stones, Beginning Linux Programing(4e), Wrox, 2007
5. Andrew S Tanenbaum, Herbert Bos, *Modern Operating systems*. Pearson Education India, 2018

CSE 3124: COMPUTER NETWORKS [2 1 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Interpret the knowledge of how computer network is structured

2. Infer the principles of different application layer protocols
3. Analyze the features in the transport layer protocols in the TCP/IP architecture.
4. Assess the role and working of network layer in TCP/IP architecture
5. Appraise the role of data link layer and the working of wireless, mobile networks.

Introduction, The Network Edge, The Network Core, Delay, Loss, and Throughput in Packet Switched Networks, Protocol Layers and their Service Models. Principles of Network Applications, The Web and HTTP, DNS, Peer-to-Peer Applications, Video streaming and Content Distribution Networks Socket Programming. Introduction and Transport-Layer Services, Connectionless Transport: UDP, Principles of Reliable Data Transfer . Connection Oriented Transport: TCP, TCP Congestion Control. Overview of Network layer, The Internet Protocol (IP): Forwarding and Addressing in the Internet, Datagram Format, IPv4 Addressing,IPV6 ,Generalized Forwarding and SDN ,Routing Algorithms- Routing in the Internet –Intra-AS Routing in the Internet: RIP, Intra-AS Routing in the Internet. The SDN Control plane , ICMP: Internet Control Message Protocol . Introduction to the Link Layer, Error-Detection and -Correction Techniques, Multiple Access Links and Protocols, Layer Addressing and ARP, Ethernet, Link-Layer Switches, Virtual Local Area Networks(VLANs). Wireless Links and Network Characteristics, The 802.11 Architecture, The 802.11 MAC Protocol, Cellular Internet Access, 3G Cellular Data Networks: Extending the Internet to Cellular Subscribers, Mobile IP

References:

1. James F. Kurose & Keith W. Ross, *Computer Networking A Top-Down Approach*, (8e), Pearson Education,2021
2. Larry L. Peterson and Bruce S. Davie, *Computer Networks- A Systems approach*, (6e), Elsevier-2021
3. Behrouz A. Forouzan, Firouz Mosharraf, *Computer Networks A top Down Approach* , Mc-Graw Hill, 2012
4. Andrew S. Tanenbaum & David J. Wetherall, *Computer Networks*, (6e), Pearson Education,2018

CSE 3141: SOFTWARE ENGINEERING LAB [0 0 3 1]

Course outcomes:

After studying this course, students will be able to:

1. Choose software design using the software modelling tool.
2. Demonstrate the implementation of Software Development Life Cycle Models for a given problem.

3. Identify any UML diagram for a given problem, to prepare ready-to-code design, and to apply test cases, to find validity of software.

Rational Rose, Star UML, SRS, SRS of Case Study, Code Generation, Project Implementation, Use Cases, SOLID Principles, Class Diagrams, Interaction Diagrams, Package Diagrams, Component Diagrams, Deployment Diagrams, NUnit, Junit.

Reference:

1. Rajib Mall, *Fundamentals of Software Engineering* (5e), PHI Learning, 2019

CSE 3142: OPERATING SYSTEMS LAB [0 0 3 1]

Course outcomes:

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

1. Demonstrate Linux system calls on files and directories.
2. Implement thread programming and inter process communication techniques.
3. Experiment with process synchronization and memory management algorithms.

Working with regular files, working with directory structures, processes and signals, file system, ipc-1:pipes and fifo, ipc-2:message queue and shared memory, deadlock , locking and synchronization, programs on threads , memory management, disk management, Tiny OS – Case Study.

References:

1. A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, (9e), Wiley and Sons (Asia) Pte Ltd, 2013
2. D.M. Dhamdhere, *Operating Systems A Concept based Approach,(2e),* Tata McGraw-Hill 2006
3. Deitel & Deitel , *Operating systems, (3e),* Pearson Education, India. 2018.
4. Neil Matthew, Richard Stones, *Beginning Linux Programming, (4e*),Wiley Publication, 2007
5. Andrew S Tanenbaum, Herbert Bos, *Modern Operating systems*. Pearson Education India, 2018
6. Sumitabha Das, *Unix concepts and applications*, (4e), McGraw Hill Education, 2017

CSE 3143: COMPUTER NETWORKS LAB [0 0 3 1]

Course outcomes:

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

1. Develop Network Application Programs.
2. Capture and analyze network traffic using network monitoring tools.
3. Design and analyze the performance of different networks and protocols.

Experiments with Wireshark : Using Wireshark and most of its basic functionalities: capturing some network traffic that is flowing through your machine now and analysing already captured network traffic. UDP and TCP Socket Programming**.** Experiments with Simulation Tools like GNS3/NS2.

References:

1. W. Richard Stevens, *UNIX Network Programming, Volume 1: The Sockets Networking API*, (3e), Addison-Wesley Professional Computing, 2003
2. Prof. Dayanand Ambawade,Deven N.Shah & Kogent Learning Solutions Inc, *Linux Lab: Hands on Linux*, Dreamtech Press, 2009
3. GNS3 Documentation, https:///www.gns3.com
4. Todd Lammle, *CCNA Cisco Certified Network Associate Study Guide*, (7e), Wiley India Pvt. Ltd 2011.

SIXTH SEMESTER

HUM 3022: ENGINEERING ECONOMICS AND FINANCIAL MANAGEMENT [3 0 0 3]

Time value of money, Interest factors for discrete compounding, Nominal & effective interest rates, Present and future worth of single, Uniform gradient cash flow. Bases for comparison of alternatives, Present worth amount, Capitalized equivalent amount, Annual equivalent amount, Future worth amount, Capital recovery with return, Rate of return method, Incremental approach for economic analysis of alternatives, Replacement analysis. Break even analysis for single product and multi product firms, Break even analysis for evaluation of investment alternatives. Physical & functional depreciation, Straight line depreciation, declining balance method of depreciation, Sum-of-the-years digits method of depreciation, sinking fund and service output methods, Costing and its types – Job costing and Process costing, Introduction to balance sheet and profit & loss statement. Ratio analysis - Financial ratios such as liquidity ratios, Leverage ratios, Turn over ratios, and profitability ratios. Safety and Risk, Assessment of Safety and Risk. Risk Benefit Analysis and Reducing Risk.

References:

1. Chan S. Park, *Contemporary Engineering Economics*, 4th Edition, Pearson Prentice Hall, 2007.

2. Thuesen G. J, *Engineering Economics*, Prentice Hall of India, New Delhi, 2005.
3. Blank Leland T. and Tarquin Anthony J., *Engineering Economy*, McGraw Hill, Delhi, 2002.
4. Prasanna Chandra, *Fundamentals of Financial Management*, Tata McGraw Hill, Delhi, 2006.
5. Mike W. Martin and Roland Schinzinger, *Ethics in Engineering*, Tata McGraw Hill, New Delhi, 2003.
6. Govindarajan M, Natarajan S, Senthil Kumar V. S, *Engineering Ethics*, Prentice Hall of India, New Delhi, 2004.

CSE 3221: PARALLEL COMPUTER ARCHITECTURE AND PROGRAMMING

[2 1 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Explain the concepts of parallel computer architectures and programming models.
2. Develop MPI programs using point-to-point and collective communication primitives.
4. Solve parallel programming tasks using OpenCL and CUDA
5. Apply CUDA programming concepts for different parallel patterns.
6. Design optimized parallel solutions using thread and memory organization in CUDA.

Introduction to Parallel processing, Parallel Computer Structures, Architectural Classification Schemes, Architecture of a modern GPU, Message passing model, MPI basic data types and functions, Point-to-point communication, Collective communication, OpenCL specification, Kernels and OpenCL execution model, OpenCL APIs, CUDA Program Structure, Vector-Vector addition, Device global memory and data transfer, Kernel functions and Threads, 1D Parallel Convolution, Atomic and Arithmetic functions, Constant Memory and caching, Parallel SPVM using CSR, CUDA Thread Organization, Matrix-Matrix multiplication, Importance of memory access efficiency, CUDA device memory types, Synchronization and transparent scalability, A strategy for reducing global memory traffic, A tiled matrix-matrix multiplication kernel, Parallel image processing applications

References:

1. D. Kirk and W. Hwu , "*Programming Massively Parallel Processors –A Hands-on approach*", Elsevier Inc.,2nd Edition, 2013.
2. Michael J. Quinn, "*Parallel Programming in C with MPI and OpenMP*", McGraw Hill Edition, 2003.

3. Benedict R. Gaster, Lee Howes, David R, Perhaad Mistry, Dana Schaa, "*Heterogeneous Computing with OpenCL*", Elsevier Inc., 1st Edition, 2012.
4. Kai Hwang and Faye A. Briggs, *Computer Architecture and Parallel Processing*, (2e), TMH Private Ltd., 2012.
5. Gonzalez, Rafael C., and Richard E. Woods. "*Digital image processing"*, Publishing house of electronics industry141.7 (2002).
6. V.Rajaraman, C. Siva Ram Murthy, "*Parallel Computers Architecture and Programming*" Prentice-Hall India, 2000.
7. Shane Cook, "*CUDA Programming: A developer's guide to parallel computing with GPUs*", Morgan Kaufman Publication, Elsevier, 2013

CSE 3222: COMPILER DESIGN [2 1 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Explain the phases of compiler and scanning techniques.
2. Construct parsers by applying different parsing techniques.
3. Identify ambiguous grammar and analyze Syntax Directed Translation techniques.
4. Construct intermediate code representations, generate and optimize the code.
5. Outline the various compiler construction tools.

Introduction, Language Processors, The Structure of a Compiler, Lexical Analysis: Role of the Lexical Analyzer, Input Buffering, Specifications and Recognition of Tokens, Design of Lexical Analyzer Generator, LEX Syntax Analysis: Introduction, Writing a Grammar, Parser Generator YACC, Top Down Parsing, Bottom Up Parsing, Introduction to LR parsing, More powerful LR parsers, Syntax-Directed Translation: Syntax-Directed Definitions, Application of Syntax- Directed Translation, Intermediate Code Generation: Variants of Syntax Trees, Three Address Code, Types and Declarations, Code Generation: Issues in Design of Code Generator, The Target Language, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, Peephole Optimization, Register allocation and Assignment.

References:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, *Compilers Principles, Techniques and Tools, (2e),* Pearson Education, 2010
2. John R. Levine, Tony Manson, Doug Brown, *LEX & YACC, (2e),* O Reilly Media, 2012.
3. Kenneth C. Louden, *Compiler Construction - Principles and Practice, (1e),* Thomson, 2007.

4. Allen L. Holub, *Compiler design in 'C', (2e),* Prentice hall,1990.
5. Vinu V. Das, *Compiler Design using FLEX and YACC,* Prentice-Hall, 2007

CSE 3241: PARALLEL PROGRAMMING LAB [0 0 3 1]

Course outcomes:

At the end of this course, students will gain the ability to

1. Implement MPI programs using point-to-point and collective communication primitives.
2. Develop CUDA programs for different parallel applications
3. Build optimized parallel solutions using thread and memory organization in CUDA

Basics of MPI, Point to Point communications in MPI, Collective communications in MPI, Error Handling in MPI, CUDA Programs on arrays, matrices, strings, different parallel patterns, image processing applications, CUDA Programs using different CUDA device memory types and synchronization.

References:

1. D. Kirk and W. Hwu , "*Programming Massively Parallel Processors –A Hands-on approach*", Elsevier Inc.,2nd Edition, 2013.
2. Michael J. Quinn, "*Parallel Programming in C with MPI and OpenMP*", McGraw Hill Edition, 2003.
3. Gonzalez, Rafael C., and Richard E. Woods. "*Digital image processing*" Publishing house of electronics industry141.7 (2002).

CSE 3242: COMPILER DESIGN LAB [0 0 3 1]

Course outcomes:

At the end of this course, students will be able to

1. Create lexical analyser and parser using C programs
2. Construct lexical analyser and parser from automation tools
3. Design an intermediate code generator and code generator

Implement all the phases of a Mini compiler for a Language under Linux environment using compiler construction tools. This includes Preliminary Scanning Applications, Identification of Tokens in a given Program, Design & Implementation of Lexical Analyzer using FLEX, Design & Implementation of Parser using BISON, Generation of intermediate code and code generation.

References:

1. Allen L. Holub, *Compiler design in 'C',* (2e), Prentice hall, 1990
2. John R. Levine, Tony Manson, Doug Brown, *LEX & YACC,* (2e), O Reilly Media, 2012
3. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, *Compilers Principles, Techniques and Tools,* (2e), Pearson Education, 2013
4. Kenneth C. Louden, *Compiler Construction - Principles and Practice,* (1e), Thomson, 2007

CSE 3243: WEB PROGRAMMING LAB [1 0 2 1]

Course outcomes:

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

1. Develop a basic website using a modern web development tool.
2. Build websites with better esthetics.
3. Create real-world web applications that interact with database and ReST APIs**.**

Basics of jQuery, Different bootstrap elements, Python programming language and its functions, Python Objects and Classes, developing a Web Application using Django, Form Processing using Django, session management techniques, accessing database using Django, Create and access ReST API.

References:

1. Mark Lutz, *Learning Python*, 5th Edition, O'Reilly, 2013
2. Nigel George, *Mastering Django*, Packt Publishing, 2016.
3. Leif Azzopardi and David Maxwell, *Tango with Django 2*, Apress, 2

SEVENTH SEMESTER

MINOR SPECIALIZATIONS

I. Computer Graphics & Visualization

CSE 4401: Digital Image Processing
CSE 4402 Computer Graphics
CSE 4403: Computer Vision
CSE 4404: Augmented and Virtual Reality

CSE 4401: DIGITAL IMAGE PROCESSING [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Summarize the fundamental concepts of image acquisition and a digital image processing systems.
2. Choose the techniques for processing the digital images in spatial and frequency domains.
3. Apply image restoration methods for noisy images.
4. Perceive different segmentation methods and their applications for grayscale and color images.
5. Implement various morphological algorithms.

Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Elements of Visual Perception, Image Sampling and Quantization, Relationships Between Pixels, Intensity Transformations and Spatial Filtering, Histogram Processing, Spatial Filtering, Smoothing, Spatial Filters, Sharpening Spatial Filters, Highpass, Bandreject, and Bandpass Filters from Lowpass Filters, Combining Spatial Enhancement Methods, Filtering In The Frequency Domain, The Discrete Fourier Transform of One Variable, Extensions to Two Variables, properties of the 2-D DFT and IDFT, The Basics of Filtering in the Frequency Domain, Image Smoothing Using Lowpass, Frequency Domain Filters, Image Sharpening Using Highpass Filters, A Model of the Image Degradation/Restoration Process, Noise Models, Restoration in the Presence of Noise Only Spatial Filtering , Estimating the Degradation Function, Inverse Filtering, Color Fundamentals, Models, Basics of Full-Color Image Processing, Fundamentals of image compression, Morphological Image Processing preliminaries, Basic Morphological Algorithms, Grayscale Morphology, Image segmentation fundamentals, Point, Line, and Edge Detection, Thresholding, Segmentation by Region Growing, Clustering and Superpixels, Graph Cuts, Morphological Watersheds, The Use of Motion in Segmentation, discussion on libraries for image processing.

References:

1. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, (4e), Pearson India Education Services Pvt. Ltd, 2018
2. Milan Sonka, Vaclav Hlavac, Roger Boyle, Image Processing, Analysis and Machine Vision,(4e), CENGAGE Learning, 2014.
3. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, Digital Image Processing Using MATLAB, (2e), Mc Graw Hill India; 2010.
4. Gloria Bueno García, Oscar Deniz Suarez, José Luis Espinosa Aranda, Jesus Salido Tercero, Ismael Serrano Gracia, Noelia Vállez Enano, Learning Image Processing with OpenCV,(1e), Packt Publishing, 2015.

CSE 4402: COMPUTER GRAPHICS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Illustrate various computer graphics output primitive algorithms.
2. Demonstrate various transformations and 2D/3D modelling techniques.
3. Implement different projections and viewing models.
4. Apply shading, animation and visible surface detection techniques.
5. Outline the fundamental concepts of information and scientific visualization.

Introduction to computer graphics, applications, physical and synthetic images, animation, rendering, relation to computer vision and image processing, review of basic mathematical objects, OpenGL architecture, primitives and attributes, modeling of two- and three-dimensional geometric objects, color models, buffering, GLUT, interaction, events and callbacks, picking, geometric transformation, affine transformations, concatenation, matrix stacks and use of model view matrix in OpenGL, Classical three dimensional viewing, parallel and perspective projective transformations, Visibility- z-Buffer, BSP trees, Open-GL culling, hidden-surface algorithms, Light sources, illumination model, shading for polygons, line segment and polygon clipping, 3D clipping, polygonal fill, Bresenham's algorithm, texture mapping, compositing, textures in OpenGL, ray tracing, Bezier curves and surfaces, B-splines, visualization, interpolation, marching squares algorithm

References:

1. Donald Hearn and Pauline Baker, *Computer Graphics with OpenGL* (4e), Prentice Hall, 2011.
2. Edward Angel, *Interactive Computer Graphics. A Top-Down Approach Using OpenGL* (6e), Pearson Education, 2012.
3. James D. Foley, Andries van Dam, Steven K. Feiner, John Hughes, Morgan McGuire, David F. Sklar, and Kurt Akeley, Computer Graphics: Principles and Practice (3e), Pearson, 2013
4. F. S. Hill Jr. and S. M. Kelley, *Computer Graphics using OpenGL* (3e), Prentice Hall, 2006.
5. Peter Shirley and Steve Marschner, *Computer Graphics* (1e), A. K. Peters, 2010.

CSE 4403: COMPUTER VISION [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Demonstrate the concepts of image formation, color models and linear filtering.
2. Identify the mathematics behind feature detection and description methods.
3. Explain the fundamental concepts in camera calibration.
4. Classify various object tracking algorithms.
5. Build object and scene recognition from images.

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

References:

1. Richard Szeliski, *Computer Vision: Algorithms and Applications*, Springer 2011.
2. David A. Forsyth and Jean Ponce, *Computer Vision: A Modern Approach*, PHI learning 2009.

CSE 4404: AUGMENTED AND VIRTUAL REALITY [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the fundamental principles of Augmented and Virtual Reality (AVR).
2. Compare the technology of displays and interfaces used in AVR.
3. Explain the techniques of multimodal user interaction and perception.
4. Apply the concepts of design and modeling used in Virtual Reality.
5. Design the applications and use cases of AVR.

Introducing AR and VR, Looking at some other types of AR and VR, Quick history tour, Evaluating the technology hype, The five Classic Components of a VR System, Displays in augmented reality, Visual Perception, Requirements and Characteristics, Three-Dimensional Position Trackers, Navigation and Manipulation Interfaces, Gesture Interfaces, Tracking, calibration and registration, Characteristics of Tracking technology, Mobile Sensors, Optical tracking, Marker Tracking, Multiple-Camera Infrared Tracking, Rendering Pipeline, Graphics and Haptics Rendering Pipeline, Geometric Modeling, AR/VR Applications- Exploring use cases, Building applications using Unity.

References:

1. Paul Mealy, *Virtual and Augmented Reality Dummies*, John Wiley & Sons, Inc., 2018
2. Dieter Schmalstieg, Tobias Hollerer , *Augmented Reality: Principles & Practice*, (1e), Addison-Wesley, 2016.
3. Burdea, G. C. and P. Coffet. *Virtual Reality Technology, (2e)*, Wiley-IEEE Press, 2006

II. Computational Intelligence

CSE 4405: Artificial Intelligence
CSE 4406: Soft Computing Paradigms
CSE 4403: Computer Vision
CSE 4408: Machine Learning

CSE 4405: ARTIFICIAL INTELLIGENCE [3 0 0 3]

Course outcomes:

The student after undergoing this course will be able to:

1. Exhibit strong familiarity with a number of important AI techniques.
2. Interpret the modern view of AI as the study of agents based intelligent systems
3. Build awareness of AI challenges in problem solving
4. Assess AI techniques and apply them to real world problems.
5. Develop self-learning and research skills to tackle a topic of interest

What is AI?, foundations of artificial intelligence, history of AI, the state of the art, agents and environment, rationality, rational agent, structure of intelligent agents, problem solving and search techniques: state space representation, production systems, uninformed search strategies, heuristic search strategies: adversarial search, constraint satisfaction problem, means-end analysis, logical agents: knowledge based agents, the wumpus world , propositional and predicate logic: representing ISA relationship, knowledge representation: ontological engineering, knowledge representation using predicate calculus, knowledge engineering process, forward and backward reasoning.

References:

1. Stuart Russell and Peter Norvig – *Artificial Intelligence* A *Modern Approach*, Pearson Education, Third Edition, 2016.
2. Elaine Rich, Kevin Knight, Shivashankar B. Nair, *Artificial Intelligence*, Third Edition, Tata McGraw Hill Edition, 2010.
3. Saroj Kaushik– *Artificial Intelligence*, Cengage Learning Publications, First Edition, 2011.
4. Don W. Patterson - *Introduction to Artificial Intelligence and Expert Systems*, PHI Publication,2006.

CSE 4406: SOFT COMPUTING PARADIGMS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Compare different neural network architectures and learning mechanisms.
2. Build supervised neural network models and their applications.
3. Demonstrate unsupervised and associative neural network models.
4. Experiment with fuzzy logic concepts and its applications.
5. Construct genetic algorithms for different applications.

Models of a Neuron, Learning Processes, Learning Tasks, Perceptron, Perceptron Convergence Theorem, Batch Learning and On-Line Learning, the Back-Propagation Algorithm, XOR Problem, Two Basic Feature-Mapping Models, Self-Organizing Map, Properties of the Feature Map, Memory matrix, Recall, Hopfield Networks, Bidirectional Associative Memory, Crisp sets and fuzzy sets, crisp and fuzzy relations, Fuzzification, Defuzzification, Fuzzy inference system, Terminologies in Genetic algorithms (GA), Simple GA, Classification of GA, Optimization of Traveling salesman problem using GA. Trees.

References:

1. Simon S. Haykin, Neural Networks and Learning Machines, Pearson Education, Third Edition, 2009.
2. S.N.Sivanandam, S.N.Deepa, Principles of Soft Computing, (2e), Wiley Publication, 2010.
3.Timothy J.Ross, Fuzzy Logic With Engineering Applications, Wiley publication, 2010.
4. M. T. Hagan, H. B. Demuth, M. Beale, Neural network design, Cengage Learning, India edition, 2010.

CSE 4403: COMPUTER VISION [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Demonstrate the concepts of image formation, color models and linear filtering.
2. Identify the mathematics behind feature detection and description methods.
3. Explain the fundamental concepts in camera calibration.
4. Classify various object tracking algorithms.
5. Build object and scene recognition from images.

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

References:

1. Richard Szeliski, *Computer Vision: Algorithms and Applications*, Springer 2011.
2. David A. Forsyth and Jean Ponce, *Computer Vision: A Modern Approach,* PHI learning 2009.

CSE 4408: MACHINE LEARNING [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the basic concepts of machine learning and supervised learning
2. Apply statistical methods for classification and inference
3. Make use of support vector machines for classification and regression
4. Apply unsupervised learning methods for clustering
5. Build decision trees for classification

Towards Intelligent Machines, Well-Posed Machine Learning Problems, Examples of Applications, Forms of Learning, Supervised Learning basics, Learning from Observations, Bias and Variance, Computational Learning Theory, Occam's Razor Principle, Metrics, Design Cycle and Issues, Statistical Learning, Bayesian Reasoning, k-NN Classifier, Discriminant Functions and Regression Functions, Linear Regression,

Logistic Regression, Parametric Methods, Support Vector Machines, Introduction, Linear Discriminant Functions, Perceptron Algorithm, Linear Maximal Margin Classifier, Linear Soft Margin Classifier, Regression, Data Clustering, Unsupervised Learning, Different Clustering Methods, Decision Tress, Introduction, Classification Decision Tree, Measure of Impurity for Evaluating Splits, Different Decision Trees, Pruning, Strengths and Weaknesses of Decision Tree Approach, Combining Multiple Learners.

References:

1. M. Gopal, *Applied Machine Learning*, McGraw Hill Education, 2018
2. Ethem Alpaydin, *Introduction to Machine Learning*, 2nd edition, MIT Press. 2010.
3. Peter Harrington, *Machine Learning in Action*, Manning Publications, 2012.
4. Andreas C. Müller & Sarah Guido, *Introduction to Machine Learning with Python*, O'Reilly Media Inc., 2017
5. Tom M Mitchell, *Machine Learning*, McGraw Hill, 2017

III. Internet of Things

CSE 4409: Introduction to IoT
CSE 4410: IoT in Agriculture
CSE 4411: IoT for Healthcare
CSE 4412: Smart Cities

CSE 4409: INTRODUCTION TO IOT [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1 Summarize the concept of IoT
2 Analyze various protocols for IoT
3 Design an IoT system using Rasperry Pi/Arduino
4 Apply data analytics and use cloud offerings related to IoT
5 Develop applications of IoT in real time scenario

Fundamentals of IoT & Smart Objects: Evolution, Impact, Challenges & Network Architecture of IoT, Core Stack, Sensor networks, IoT Architectures Related Protocols: Physical and MAC layers, Network Layer, Optimizing IP for IoT, Routing over Low Power and Lossy Networks, Supervisory Control and Data Acquisition, Application Layer Protocols, IoT Communication Protocols: Wi-Fi, ZigBee, Bluetooth, Edge Computing: Basic Architecture, Design And Development of IoT: Microcontroller, System on Chips, Building Blocks, Arduino Board, Raspberry Pi - Interfaces and Python Programming, Data Analytics And Supporting Services: Machine Learning, No SQL

Databases, Hadoop Ecosystem, Apache Kafka & Spark, Edge Streaming and Network Analytics, Cloud for IoT, Web Application Framework, System Management, Case Studies / Industrial Applications

References:

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, and Jerome Henry, *IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things*, Cisco Press, 2017.
2. Arshdeep Bahga, Vijay Madisetti, *Internet of Things – A Hands-on Approach*, Universities Press, 2015.
3. Rajkumar Buyya, Amir Vahid Dastjerdi, *Internet of Things – Principle and Paradigms*, Elsevier, 2016.
4. Olivier Hersent, David Boswarthick, Omar Elloumi, *The Internet of Things – Key Applications and Protocols*, Wiley, 2012.
5. Khaled Salah Mohamed, "*The Era of Internet of Things Towards a Smart World*", Springer, 2019.

CSE 4410: IOT IN AGRICULTURE [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the basic concepts in agricultural IoT
2. Analyze the requirements for the development of an agricultural IoT application
3. Model and design the agricultural IoT application project
4. Implement the agricultural IoT application project
5. Make use of standards in agricultural IoT application

Introduction to Agricultural IoT: an integrated view on precision smart farming from a multidisciplinary perspective, Agricultural internet of things and decision support for precision smart farming: Challenges and Developments in Sustainable Agriculture, IoT-Enabled Agricultural System Applications, Challenges and Security Issues, A Design of IoT-Based Agricultural System for Optimal Management: Data Mining Techniques and Their Role in IoT, Classifications, Decision Tree, Processing Agriculture Data Using Other Techniques, Adopting Big Data Analysis in the Agricultural Sector: Financial and Societal Impacts, AI-Based Yield Prediction and Smart Irrigation, Case studies in IoT-based Precision agriculture

References:

1. Annamaria Castrignanò, Gabriele Buttafuoco, Raj Khosla, Abdul M. Mouazen, Dimitrios Moshou, Olivier Naud, *Agricultural Internet Of Things And Decision Support For Precision Smart Farming*, Elsevier Publications 2020.
2. Prasant Kumar Pattnaik, Raghvendra Kumar, Souvik Pal, *Internet of Things and Analytics for Agriculture, Volume 1* , Edited, Springer Series 2020.
3. Dimitrios Serpanos, Marilyn Wolf, *Internet-of-Things (IoT) Systems: Architectures, Algorithms, Methodologies*, Springer, 2020
4. D. Kent Shannon, David E. Clay, and Newell R, *Precision Agriculture Basics*, Kitchen American Society of Agronomy, Crop Science Society of America, Soil Science Society of America, 2020

CSE 4411: IOT FOR HEALTHCARE [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Apply concepts of Embedded Systems for IOT
2. Analyse the communication protocols for communication among IOT devices
3. Make use of Internet Technologies in IOT
4. Design the Healthcare systems with IOT
5. Summarize real time applications of IOT

IOT: Embedded Systems-an overview, Features, Networked Embedded System types, Wireless communication standards-zigbee, Bluetooth & Wi-Fi. OSI & TCP/IP models. Introduction to the Internet in IOT. Introduction to Smart Objects or Things. Application in health-care systems- Patient Monitoring & diagnostics, Home healthcare & Personal care & Fitness. IOT Hardware Platform& Sensor Interface. Energia Wi-Fi libraries. Sensor interface: Temperature sensor, pressure sensor, Light sensor, IR sensor. Client-Server Communication Paradigm Basic Client-Server communication model, Network Sockets, Ports, Examples of client server communication, Energia client & server class APIs. Embedded Web-Server & IOT Cloud Services Embedded web server. Design of a simple embedded web server. Health Care Monitoring Systems, Case Studies.

References:

1. Cuno P Fister, *Getting Started with Internet of Things*, O'Reily Publishers, 2011.
2. J. P Vasseur, Adam Dunkels, *Interconnecting Smart Objects with IP*, CISCO publishers, 2010.
3. Raj P, Chatterjee J M, Kumar A, Balmurugan B, *Internet of Things Use Cases for the Healthcare Industry*, Springer Publishers, 2020.

4. Gupta Nishu, Paiva Sara, *IOT and ICT for Healthcare Applications*, Springer Publishers, 2020.

CSE 4412: SMART CITIES [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Design solutions for challenges in implementing IoT for smart city
2. Analyse the requirements for IoT application with respect to Smart Cities
3. Implement the IoT for improving urban infrastructure
4. Analyse the data acquired through IoT smart city application
5. Design secured smart city IoT application

Why Smart Cities?, Impact of Internet of Things in smart cities, IoT-based smart water management, Design of smart urban drainage systems using evolutionary decision tree model , Key points in Smart cities applications , Smart City Planning and Management, Big Data Analytics Processes and Platforms Facilitating Smart Cities , Dimension Reduction for Big Data Analytics in Internet of Things , Autonomous Radios and Open Spectrum in Smart Cities, Smart Cities Challenges: Allahabad, Amravati, Naya Raipur. Smart city projects under smart city mission in Dharmashala, Himachal Pradesh

References:

1. IoT Technologies in Smart Cities From sensors to big data, security and trust Edited by Fadi Al-Turjman and Muhammad Imran, Institution of Engineering and Technology, 2020
2. Building Smart Cities-Analytics, ICT, and Design Thinking by Carol L. Stimmel, CRC press, 2016 Edition.
3. Smart Cities -Foundations, Principles, and Applications, Wiley Publicatins, 2017
4. Internet of Things for Smart Cities-Technologies, Big Data and Security by Waleed Ejaz· Alagan Anpalagan, Springer 2018.

IV. Data Analytics

CSE 4413: Data Warehouse and Data Mining
CSE 4414: Natural Language Processing
CSE 4403: Computer Vision
CSE 4416: Big Data Analytics

CSE 4413: DATA WAREHOUSE AND DATA MINING
[3 0 0 3]

Course outcomes:

After studying this course, students will have the ability to:

1. Design and implement a data warehouse, and extract aggregated data for data mining activities
2. Apply suitable data preprocessing methods to make the raw data suitable for data mining
3. Discover association rules from real world data
4. Build and apply classifiers to real world data
5. Build and apply clustering techniques to real world data

Introduction to data mining, Kinds of data, Data Warehouse Basic Concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, Data Generalization by Attribute Oriented Induction, Data Preprocessing, Frequent Itemset Generation, Rule Generation, FP-Growth Algorithm, Evaluation of Association Patterns, Decision Tree Induction, Model Overfitting, Evaluating the Performance of a Classsifier. Rule-Based Classifier, Nearest-Neighbour classifiers, Bayesian Classifiers, Artificial Nerual Network, Support Vector Machine, Ensemble Methods, Class Imbalance Problem, Multiclass problem, K-Means, Agglomerative Hierarchical Clustering, DBSCAN, Density-Based Clustering, Graph-Based Clustering, Outlier Detection, Clustering-based Techniques

References:

1. Jiawei Han and Micheline Kamber, *Data Mining- Concepts and Techniques*,(3e), Morgan Kaufmann Publishers, 2011
2. Pang-Ning-Tan, Michael Steinbach, Vipin Kumar, *Introduction to Data Mining*, (2e), Pearson ,2018
3. Ian H Witten, Eibe Grank, Mark A Hall, *Data Mining, Practical Machine Learning Tools and Techniques*, (4e), Morgan Kaufmann Publishers, 2017
4. Parteek Bhatia, *Data Mining and Data Warehousing: Principles and Practical Techniques*, Cambridge University Press, 2019
5. Paulraj Ponniah, *Data Warehousing*, (2e), Wiley India Pvt. Ltd., 2010

CSE 4414: NATURAL LANGUAGE PROCESSING [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the basic concepts and techniques of Natural Language Processing
2. Explain the various language modelling and smoothing techniques
3. Identify information from text using POS tagging of English language
4. Apply Context-Free Grammar for Natural Language Processing
5. Discuss the elements and applications of parsing.

Knowledge in Speech and Language Processing, Ambiguity, Models and Algorithm. Survey of English Morphology, Finite-State Morphological Parsing, Building a Finite-State Lexicon, FSTs for Morphological Parsing, Lexicon-Free FSTs. Words and sentence tokenization, Detecting and Correcting Spelling Errors. Case study: Normalizing Text, Segmentation. N-Grams, Unsmoothed N-Grams, Smoothing, Interpolation, and Back-off. English Word Classes, Tag-sets for English, Part-of-Speech Tagging, The Noisy Channel Model for Spelling. Case study: Automatic Tagging. Constituency, Some Grammar Rules for English, The Penn Treebank project, Dependency Grammar. Parsing with Context Free Grammars, CKY algorithm, Statistical Parsing.

References:

1. Daniel Jurafsky & James H. Martin, *Speech and Language Processing*, (2e), Pearson, 2009.
2. Steven Bird, Ewan Klein and Edward Loper, *Natural Language Processing with Python*, (1e), O'Reilly Media, 2009
3. Akshar Bharati, Rajeev Sangal and Vineet Chaitanya, *Natural Language Processing: A Paninian Perspective*, Prentice-Hall of India, New Delhi, 1995
4. Steven Bird, Ewan Klein, Edward Loper, *Natural Language Processing with Python – Analysing Text with natural language toolkit* , O'Reilly Media, 2009
5. Chris Manning, Hinrich Schutze, *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, 1999.

CSE 4403: COMPUTER VISION [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Demonstrate the concepts of image formation, color models and linear filtering.
2. Identify the mathematics behind feature detection and description methods.
3. Explain the fundamental concepts in camera calibration.
4. Classify various object tracking algorithms.
5. Build object and scene recognition from images

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

References:

1. Richard Szeliski, *Computer Vision: Algorithms and Applications*, Springer 2011.
2. David A. Forsyth and Jean Ponce, *Computer Vision: A Modern Approach*, PHI learning 2009.

CSE 4416: BIG DATA ANALYTICS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Examine the issues that occur while analyzing big data and identify the tools to be used.
2. Develop a recommendation system for a real-world problem
3. Classify Big Data by applying algorithms based on supervised and unsupervised techniques.
4. Analyse graph data using GraphX.
5. Apply Monte-Carlo simulation for estimating risk and analyse data streams using Spark streaming.

Introduction to Big Data, Hadoop, Map Reduce, Yarn architecture, Hive and Apache Spark, Anatomy of Spark application, Spark Manager and Cluster Manager, Spark Applications, RDD, The alternating least squares recommender algorithm, Evaluating recommendation Hyperparameter selection, Fast forward to regression, Decision trees

and forests, Decision tree hyperparameters, Tuning decision trees, Random decision forests, Anomaly detection, K-means clustering, Network intrusion, visualization, Clustering in Action, Analyzing the MeSH major topics and their co-occurrences, Constructing a co-occurrence network with GraphX, Understanding the structure of networks, Monte-Carlo simulation, Pre-processing, Determining the factor weights, Visualizing the distribution of returns, Evaluating the results, Spark streaming architecture, DStreams, Apache Kafka

References:

1. Vignesh Prajapathi, *Big Data Analytics with R and Hadoop,* Packt Publishing, 2013.
2. Jeffery Aven, *Data Analytics with Spark using Python,* Pearson, 2018
3. Sandya Ryza, Uri Laserson, Sean Owen and Josh Wills, *Advanced Analytics with Spark (2e),* O'Reilly Media Inc, 2017
4. Holden Karau, Andy Konwinski, Patrick Wendell and Matei Zaharia, *Learning Spark: Lightning-Fast Big Data Analysis (2e)*, O'Reilly Media Inc, 2020.
5. Tom White, *Hadoop: The definitive guide (4e)*, O'Reilly, 2015.

V. Cyber Security

CSE 4417: Network Security

CSE 4418: Cyber forensics

CSE 4419: Artificial intelligence in Cyber security

CSE 4420: Database and Application security

CSE 4417: NETWORK SECURITY [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the concepts of computer and network security
2. Identify the methods to protect the system against intruders and malicious software
3. Compare the techniques used for providing security at the transport and network levels
4. Analyse the methods for providing network access control and cloud security
5. Explain the methods used for providing wireless network security

Computer and Network Security Concepts, The OSI Security Architecture, Security Attacks, Service, Mechanisms, Model for Network Security, System security, Malicious Software Viruses, Worms, Trojans, Zombie, Bots, Keyloggers, Phishing, Spyware, Intruders, Intrusion Detection, Password Management, DDoS Attacks, Firewalls-Need, Types,Configurations and functions, Transport Layer Security, HTTPS, SSH, IP Security, Policy, Encapsulating Security Payload, , IKE, Network Access Control, Extensible Authentication Protocol, IEEE 802.1X Port-Based Network Access Control, Cloud Security Risks and Countermeasures, Data Protection in the Cloud, Wireless Security, Mobile Device Security, IEEE 802.11i Wireless LAN Security

References:

1. William Stallings, *Network Security Essentials: Applications and Standards ,(6e),* Pearson Education Limited, 2018
2. William Stallings, *Cryptography and Network Security: Principles and practice, (7e),* Pearson Education Limited, 2017
3. Behrouz A. Forouzan, Debdeep Mukhopadhyay, *Cryptography and Network Security, (3e),* McGraw Hill Education, 2015
4. Atul Kahate, *Cryptography and Network Security, (4e),* McGraw Hill, 2019
5. John R. Vacca, *Computer and Information Security Handbook, (3e),* Morgan Kaufmann, 2017

CSE 4418: CYBER FORENSICS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Define Software and Hardware for Cyber Forensics
2. Classify Cyber Crimes.
3. Apply Cyber Laws
4. Categorise Cyber Investigation.
5. Estimate Financing in Cyber Forensics.

Networks and Network security, Network Models, Network Topologies, Network Layer, Network Protocols, Introduction to Cybercrime, Types of Cyber Crime, Classification of Cyber Crime, Cyber Crime (Present and Future), Introduction to Cyber Forensics, Disk Forensics, Digital Evidence: Acquisition, Analysis, Admissibility, Malware Forensics, Email Forensics, Memory Forensics, Acquisition, Cyber Crime Case Studies, Cyber Crime against individuals, Cyber Crime against Nation. Cyber Laws, Indian Cyber Laws, International Cyber Laws.

Text Book:

1: Dejey, S Murugan(2018)- Cyber Forensics, Oxford University Press, New Delhi.

References:

1. Augustine T. Paul, *Combating Cyber Crime*, Crescent Publishing Corporation, New Delhi, 2007
2. Augustine T. Paul, *Cyber Crime and Legal Issues*, Crescent Publishing Corporation, New Delhi, 2007
3. Augustine T. Paul, *Intellectual Property crime*, Crescent Publishing Corporation, New Delhi, 2007
4. Barkha & Mohan U.Rama, *Cyber Law & Crimes IT Act 2000 and Computer Crime Analysis*, Asia Law House, Hyderabad.
5. Bryan, Kellie, Dunnesen, Kristen & Jean Jayson, *Cyber Fraud (Tactics, Techniques and Procedures)*, CRC Press, Taylor & Francis, London, 2009

CSE 4419: ARTIFICIAL INTELLIGENCE IN CYBER SECURITY [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Summarize the concepts of intelligent agents used in AI.
2. Select mechanisms to reach the goal state using searching techniques.
3. Develop the necessary axioms from the existing knowledge base in logic.
4. Apply machine learning algorithms for preventing poisoning and evasion attacks.
5. Demonstrate the usage of machine learning algorithms for network intrusion detection.

Foundations of Artificial Intelligence, History of Artificial Intelligence, The state of the Art, Agents and Environments, The concept of Rationality, The Nature of Environments, The structure of Agents, Problem Solving agents, Searching for Solutions, Uninformed search strategies, Informed (Heuristic) search strategies, Ontological Engineering, Categories and objects, Reasoning systems for categories, Introduction to Knowledge Engineering in Cybersecurity, Cybersecurity Taxonomies, Upper Ontologies, Machine Learning Algorithms Are Vulnerable, Threat Model, Data Poisoning, Computing Evasion Attacks, Network Intrusion Detection Systems-Deployment Methods, Detection Methodologies. Machine Learning in Network

Intrusion Detection - Artificial Neural Networks, Deployment of ML-Based NIDSes. Experiment-Evaluation.

References:

1. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach (3e*), Pearson 2010.
2. Leslie F. Sikos (Ed), *AI in Cyber Security*, Intelligent Systems Reference Library, Volume 151, Springer, 2019, ISBN 978-3-319-98841-2.
3. Elaine Rich, Kevin Knight, Shivashankar B. Nair, *Artificial Intelligence (3e*), Tata McGraw Hill, 2017.

CSE 4420: DATABASE AND APPLICATION SECURITY [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Summarize the fundamentals of information security.
2. Make use of administration privileges for authentication.
3. Apply security models for database applications.
4. Identify the best practices for secure database communications.
5. Develop an authentication framework to overcome various attacks.

Introduction, Database security, Operating systems overview, security environment, Authentication methods, Vulnerabilities of operating systems, Defining and using profiles, Designing and implementing password policies, Granting and revoking user privileges, Types of users, security models, application types, application security models and Data encryption, Secure database links, Monitor usage of database links, Map data sources and sinks, Understand Web services security before exposing Web services endpoints, Attacking authentication and securing authentication, Attacking data stores and how to prevent it, Attacking users using Cross-site scripting and other techniques.

References:

1. Hassan A. Afyouni, *Database Security and Auditing*, India Edition, CENGAGE Learning, 2009.
2. RonBen Natan, *Implementing Database Security and Auditing*, Elsevier, Indian Reprint, 2006.
3. Dafydd Stuttard and Marcus Pinto, The Web Application Hacker’s Handbook, Wiley Publishing, Second edition, 2011.

4. Castano S and Fugini M., *Database Security*, Addison Wesley, ACM, 2004.
5. Fernandez, Summers and Wood, *Database Security and Integrity*, Addison Wesley, 2012.

Other Programme Electives

CSE 4441: ADVANCED ALGORITHMS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Analyze different types of data structure operations and their cost finding techniques
2. Explain various operations on B-Tree and disjoint sets.
3. Identify shortest paths from single source to all other vertices using various algorithms.
4. Discuss different algorithms to find shortest paths for all pairs of vertices
5. Apply multi-threaded algorithms for suitable applications

Aggregate analysis, The accounting, potential method, Dynamic Tables, B-Trees and Basic operations on B-Trees, Binomial trees and Binomial heaps, Disjoint-set operations, Linked-list representation of disjoint sets, Disjoint set forests, The Bellman-Ford algorithm, Single-source shortest paths in directed acyclic graphs, Difference constraints and shortest paths, Shortest Paths and matrix multiplication, Johnson's algorithm for sparse graphs, Basics of dynamic multithreading, Multithreaded matrix multiplication, Multithreaded merge sort.

References:

1. Cormen Thomas H., Leiserson Charles E, Rivest Ronald L. and Stein Clifford, "Introduction to Algorithms", (3e), MIT Press, 2009.
2. Baase Sara and Gelder A.V., "Computer Algorithms -Introduction to Design and Analysis", (3e), Pearson Education, 2000
3. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", (3e), Pearson Education, 2011

CSE 4442: ANDROID APPLICATION DEVELOPMENT [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Illustrate the concepts of life cycle to schedule the sequence of events and handle design specific UI controls.
2. Perceive the process of data handling within the application.

3. Make use of the database by associating with multimedia, sensor and other devices.
4. Build the access control mechanism for users to access the application.
5. Design simple Android applications.

Mobility landscape, Mobile platforms, overview of Android platform, App user interface designing, Activity- states and life cycle, interaction amongst activities, Threads, Async task, Services, Notifications, Broadcast receivers, Telephony and SMS APIs, Native data handling, file I/O, shared preferences, mobile databases, enterprise data access, Content Providers, Graphics and animation APIs, multimedia, location based services, sensors**,** maps, Debugging mobile apps, testing Apps, test automation, packaging mobile apps, distributing apps.

References:

1. John Hortan, Android Programming with Kotlin for Beginners, Packt Publishing Ltd., 2019.
2. Anubhav Pradhan, Anil V Deshpande , *Composing Mobile Apps , learn, explore apply using Android*, (1e), Wiley India Pvt. Ltd., 2014.
3. Van Drongelen, Mike, *Android Studio Cookbook*, Packt Publishing Ltd, 2015.
4. Lee, Wei-Meng, *Beginning android 4 application Development*, John Wiley & Sons, 2012.
5. https://developer.android.com/guide/index.html
6. Meier, Reto, *Professional Android 4 application development*, John Wiley & Sons, 2012.
7. Mednieks, Zigurd R., et al., *Programming android,* O'Reilly Media, Inc., 2012.

CSE 4443: ANIMATION TECHNOLOGIES [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:
1. Summarize the fundamentals of animation technology.
2. Illustrate various animation and motion capture techniques.
3. Make use of computer graphics methods in animation.
4. Analyze various types of multimedia animation.
5. Appraise case studies of animation technologies.

Animation, The Past, Animated Cartoons and their Evolution, Editorial cartoons, Computer animation, Limited animation, Motion capture, Multimedia and Animation, Machinima, Animatronics, and Computer Graphic and Animation, Different types of animation: 2D animation, 3D animation, Stop Motion animation, sand animation, pixilation, paint on glass animation, clay animation, puppet animation, object animation, Cut out animation, Time lapse animation, Kinestasis and collage. Studies on the films of: Walt Disney, MGM cartoon studios, Warner Bros Studios, Pixar Studio, Studio Ghibli.

References:

1. Jiang Tan, Aspects of animation, steps to learn animated cartoon, Serials Publications Pvt. Ltd, 2016
2. Richard Williams, Animation Survival Kit revised edition, Faber; Main - Revised edition 2009
3. Kit Laybourne and John Canemaker, The Animation book: A complete guide to animated film making, Three Rivers Press, 1998
4. Charles Solomon, Enchanted Drawings: The history of animation,1994
5. Bob Thomas, The Art of Aanimation,1958
6. Ken A. Priebe, The Advanced Art of Stop-Motion Animation, Course Technology, a part of Cengage Learning, 2011
7. Lisa Lee, Animation for Beginners : Basic Principles of Animation for Motion Graphics, 2019

CSE 4444: BLOCKCHAIN TECHNOLOGY [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1 Outline basic concepts and structure of blockchain technology
2 Apply the concept of asymmetric key encryption and hashing to manage the integrity of blocks in blockchain.
3 Build fault tolerant systems.
4 Analyse the working of Ethereum blockchain technology.
5 Apply the concept of smart contracts to enhance transparency and transparency among participants.

Blockchain 101: Processing a financial transaction , Ledger, Concept of a trustless system, Introducing blockchain: General elements of blockchain, Peer-to-peer network, Block, Types of blockchains. Byzantine generals problem. Consensus. Components and structure of blockchain Blocks, Example Ethereum block, Bitcoin block, Blockchain miners, Blockchain validators, Smart contracts, Blockchain speed. Cryptography and Mechanics behind blockchain Principles of security, Symmetric cryptography, Asymmetric (public-key) cryptography, Signatures, Hashing. Achieving consensus: Practical Byzantine fault tolerance algorithm, Byzantine faults, Proof of Work: Proof of Stake: Tender mint consensus. Proof of Authority, Establishing authority. Proof of Elapsed time. Ethereum: Introducing Ethereum, Components of Ethereum: Ethereum accounts, Ethereum network, Ethereum gas, Ethereum virtual machine. Smart contract: Why smart contracts? : Automating processes and resolutions between parties, Real-world example, Example Ethereum smart contracts, Limitations of smart contracts.

References:

1. Brenn Hill , Samanyu Chopra, Paul Valencourt, Blockchain Quick Reference: A guide to exploring decentralized blockchain application development, (1e), Ingram short title 2018

2. Andreas M. Antonopoulos, “Mastering Bitcoin: unlocking digital cryptocurrencies”, O'Reilly Media, (1e) 2014
3. Roger Wattenhofer, “Distributed Ledger Technology, The science of the Blockchain”, Inverted Forest Publishing, (2e), 2017.
4. Antonopoulos, Andreas M. and Wood, Gavin. “Mastering Ethereum”, O’Reilly Media, 2018.
5. George Icahn, “Blockchain the complete guide to understanding blockchain technology”, Amazon publishers, 2017.

CSE 4445: CLOUD COMPUTING [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Distinguish between conventional computing systems and cloud computing systems.
2. Discuss the role of virtualization in Infrastructure as a service
3. Illustrate the issues related to service oriented architecture
4. Demonstrate the cloud management applications using cloud programming model.
5. Identify security threats for a given cloud application.

Introduction to Cloud Computing, Virtualization and Infrastructure as a service, Hyper converged Infrastructure, Virtual Machines Provisioning and Migration Services, Services and Service Oriented Architectures, Message-Oriented Middleware, Portals and Science Gateways, Cloud Programming and Software Environments: Features of Cloud and Grid Platforms, Parallel and Distributed Programming Paradigms, SLA Management, SLA Management in Cloud, Automated Policy-based Management. Cloud Security Fundamentals, Aspects of Data Security, Data Security Mitigation, Provider Data and Its Security, Security Management Standards, Security Management in the Cloud, Availability Management, SaaS Availability Management, PaaS Availability Management, IaaS Availability Management, Access Control

References:

1. Rajkumar Buyya, James Broberg, Andrzej Goscinski, *Cloud Computing Principles and Paradigms*, Wiley Publications, 2013.
2. Matthew Portnoym, *Virtualization Essentials*, John Wiley and Sons Publication, 2012
3. Thomas Erl, *Service oriented Architecture*, Pearson publications, 2016
4. Scott D Lowe, *Hyper converged Infrastructure implementation strategies*, Actual Tech media, 2015

5. Tim Mather, Subra Kumaraswamy, Shahed Latif , *Cloud Security and Privacy*, O'Reilly Media, Inc., 2009.
6. Kai Hwang,Geoffrey Fox, JackDongarra,Todd Green, *Distributed and Cloud Computing: Clusters, Grids, Clouds and The Future Internet*, Morgan Kaufmann Publishers,2011

CSE 4446: CRYPTANALYSIS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Summarize the cryptanalysis techniques to break the classical ciphers.
2. Analyse various attacks on block and stream ciphers.
3. Compare different algorithms used to break the public key cryptosystems.
4. Apply methods to cryptanalyze the public key cryptosystem.
5. Demonstrate the attacks on hash functions.

Cryptanalysis of classical ciphers, modern block and stream ciphers, Perfectly Secret Encryption, Computationally Secure Encryption ,Chosen-Plaintext Attacks (CPA), Chosen-Ciphertext Attacks (CCA), Analysis of stream ciphers, LFSR and RC4, Differential and Linear cryptanalysis Analysis of DES and AES, Cryptanalysis of public key cryptosystems, Primality Testing, The Solovay-Strassen Algorithm, The Miller-Rabin Algorithm, Factoring algorithms, The Pollard's p-1 Algorithm, The Pollard's Rho Algorithm, Dixon's Random Squares Algorithm, Other attacks on RSA Algorithms for computing Discrete Logarithms, Generic Attacks on hash functions, Birthday attacks for finding collisions, Security of Hash Functions, Random-Oracle model.

References:

1. Jonathan Katz and Yehuda Lindell, "*Introduction to Modern Cryptography*" *(3e),* CRC Press, 2020
2. Douglas R. Stinson, "*Cryptography Theory and Practice", (4e),* CRC Press, 2018.
3. Antoine Joux, "*Algorithmic Cryptanalysis"*, CRC Press, 2009.
4. Richard Klima, Neil Sigmon, "*Cryptology Classical and Modern",(2e)* Chapman and Hall/CRC 2018.
5. Hans Delfs and Helmut Knebl, *"Introduction to Cryptography: Principles and Applications", (3e),* Springer- Verlag, 2015.

CSE 4447: DEEP LEARNING [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Explain mathematical concepts for deep learning applications
2. Demonstrate a working knowledge of neural networks and deep learning
3. Implement the concept of regularization and optimization for training deep models
4. Apply the convolutional architectures, recurrent and recursive neural network for deep models
5. Evaluate the performance of the models using standard metrics

Introduction, Mathematical Preliminaries, Machine Learning Basics: Learning, Supervised and Unsupervised learning algorithms, Deep Feedforward Networks: Hidden units, Backpropagation algorithm, Regularization for Deep Learning: Parameter Norm Penalties, , Regularization and Under-Constrained Problems, Dataset Augmentation, Noise-Robustness, Bagging and Other Ensemble Methods, Dropout, Optimization for Training. Deep Models: Challenges in Neural Network Optimization, Convolutional Networks, Recurrent and Recursive Networks, Practical Methodology: Performance Metrics, Default Baseline Models, Selecting hyper parameters, Debugging Strategies.

References:

1. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press 2016.
2. Simon Haykin, Neural Networks and Learning Machines, PHI, 2008
3. Andrew Ng's Notes on Machine Learning from CS229
4. François Chollet, *Deep Learning with Python*, Manning Publications, 2017
5. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, Mathematics for Machine Learning, Cambridge University Press,2020.

CSE 4448: DISTRIBUTED SYSTEMS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Explain the different architectures of distributed systems.

2. Illustrate interprocess communication between processes.
3. Demonstrate the working of a naming service in a distributed system.
4. Analyze synchronization and coordination between processes.
5. Classify various mechanisms for maintaining consistency in the replicated data.

Distributed system, Design Goals, Architectural Styles, Middleware Organization, System Architecture, Example Architectures, Communication, Foundations, Remote procedure call, Message-oriented communication, Multicast communication, Naming, Names, Identifiers and Addresses, Flat naming, Structured naming, Coordination, Clock synchronization, Logical clocks, Mutual exclusion, Election algorithms, Consistency & Replication, Introduction, Data-centric consistency models, Client-centric consistency models, Replica management, Consistency protocols.

References:

1. Maarten van Steen and Andrew S. Tanenbaum, *Distributed Systems (3e)*, 2017.
2. Coulouris G., Dollimore J., and Kindberg T., *Distributed Systems* (4e), Pearson, 2009.
3. Ajay D. Kshemkalyani, and Mukesk Singhal, *Distributed Computing: Principles, Algorithms, and Systems*, Cambridge University Press; Reissue edition, March 2011.
4. Mei- Ling Liu, *Distributed Computing: Principles and Application*, Pearson Education, Inc. New Delhi. 2004.

CSE 4449: ETHICAL HACKING AND CYBER SECURITY [3 0 0 3]

Course outcomes:

After studying this course, students will be able:

1. Outline the concepts of ethical hacking and cyber security.
2. Analyze tools and techniques to carry out ethical hacking.
3. Evaluate security techniques used to protect system and user data.
4. Demonstrate various steps in ethical hacking.
5. Categorize the different security attacks and prevention mechanisms.

Computer Security concepts, Introduction to ethical hacking, Phases in ethical hacking, TCP/IP concepts review, Network and computer attacks, Foot-printing and Social engineering. Port Scanning, Types of port scans, Enumerating Windows operating systems, NETBIOS enumeration tools, DumpSec, Hyena, Nessus and open VAS, Tools for identifying vulnerabilities, Built-in Windows tools, Best practices in hardening Windows systems, Patching systems, Antivirus solutions, Windows OS vulnerabilities,

Hacking web servers, Understanding web application vulnerabilities, Application vulnerabilities and counter measures, Tools for web attackers and security testers, Cybercrime: Mobile and Wireless devices, Tools and methods used in cybercrime, Cybercrimes and cybersecurity**.**

References:

1. Michael T. Simpson, Nicholas D. Antill, *Hands-On Ethical Hacking and Network Defense*, (3e), Cengage Learning, 2016.
2. William Stallings, *Cryptography and Network security*, (7e), Pearson, 2017.
3. Sumit Belapure, Nina Godbole, *Cyber Security: Understanding cybercrimes, Computer Forensics and Legal perspectives*, (1e) Wiley India, 2011.
4. Rafay Baloch, *Ethical hacking and penetration testing guide*, CRC Press, Taylor & Francis Group, 2015.
5. Kimberly Graves, *Official Certified Ethical Hacker Study Guide*, (1e), Wiley India ,2010.
6. Mark Taber, *Maximum Security: A Hacker's Guide to Protecting Your Internet Site and Network*, 1997.

CSE 4450: GAME PROGRAMMING [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the game elements and game design atoms.
2. Create balance between chance and skill in game design.
3. Illustrate the basic unity concepts and interface.
4. Create environment and characters using unity terrain editor and controller object.
5. Develop a mini-game with game physics and game mechanics.

Building Blocks of Game Design- Basics, Game Design Atoms, Chance & Skill in Games and Game Genres- Elements of Chance, Elements of Strategic Skill, Elements of Twitch Skill, Chance & Skill, Finding the Balance, Genres of Games, Unity Basics- Enter the Third Dimension: Getting to Grips with 3D, Rigid body Physics, Prototyping and Scripting Basics, Environment and Characters Creation in Unity- Creating the Environment: Using the Terrain Editor, Player Characters and further Scripting: Working with the Inspector, Anatomy of a Character, Scripting for Character Movement, Colliders and Rigid Body in Unity- Interactions: Adding the Outpost, Collisions and Triggers, Collection, Inventory and HUD: Writing the Player Inventory, Restricting Outpost Access, Displaying the Power Cell HUD, Coconut Shy Game.

References:

1. Brathwaite, Brenda, and Ian Schreiber, *Challenges for Game Designers: Non-digital Exercises for Video Game Designers*, Course Technology, Cengage Learning, 2017
2. Ernest Adams, *Fundamentals of Game Design*, (2e), New Riders, Pearson, 2010
3. Will Goldstone, *Unity 3.x Game Development Essentials: Game Development with C# and Javascript*, (2e), PACKT publishing, 2011

CSE 4451: HARDWARE SECURITY [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Analyse the state-of-the-art cryptographic hardware design techniques
2. Implement standard cryptographic methods in hardware
3. Design and Analyse secure hardware design mechanisms
4. Understand the side channel analysis through timing analysis, simple power analysis, differential power analysis and countermeasures
5. Classify various side channel attacks on cryptographic hardware design

Introduction to Modern Cryptography, Efficient and unified method for hardware cryptography. Modern Hardware Design, The Field Programmable Gate Array (FPGA) architecture, FPGA implementation of AES, Hardware architectures of symmetric block ciphers, Implementation of Basic operations of AES in hardware, Secure and efficient implementation of symmetric key encryption schemes using FPGAs, Block cipher modes of operation from a hardware implementation perspective, high speed implementation of Elliptic Curve scalar multiplication on FPGAs. Side Channel Analysis (SCA), Fault attacks, Cache attacks, Scan chain based attacks. Improved techniques for side channel analysis, Electromagnetic attacks and countermeasures.

References:

1. Debdeep Mukhopadhyay and Rajat Subhra Chakraborty, "*Hardware Security: Design, Threats and Safeguards*", CRC Press, 2015.
2. Cetin Kaya Koc, "*Cryptographic Engineering*", Springer, 2009.

3. Mohammad Tehranipoor and Cliff Wang, "*Introduction to Hardware Security and Trust*", Springer, 2012.
4. Nadia Nedjah and Luiza de Macedo Mourelle, "*Embedded Cryptographic Hardware: Design & Security*", Nova Publishers, 2005.
5. Stefan Mangard and Elisabeth Oswald and Thomas Popp, "*Power Analysis Attacks: Revealing the Secrets of Smart Cards*", Springer, 2007.

CSE 4452: HIGH PERFORMANCE COMPUTER ARCHITECTURE [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Design different SIMD interconnection networks for array processors
2. Explain dataflow computers
3. Illustrate symmetric multiprocessors, cluster architecture and multithreading concepts
4. Analyze the performance issues on multicore architectures
5. Summarize the varieties of Supercomputers

Review of SIMD Computer Organization, SIMD interconnection networks, Parallel Algorithms for Array processors, Data flow computers, Symmetric Multiprocessor Organization, Cache Coherence and the MESI Protocol, Multithreading and Chip Multiprocessors, Synchronization, Models of Memory Consistency, Clusters, Operating System Design Issues, Cluster Computer Architecture, Blade Servers, Clusters Compared to SMP, Multicore Computers, Hardware Performance Issues: Increase in Parallelism, Power Consumption, Software Performance Issues: Software on Multicore, Multicore Organization, Intel x86 Multicore Organization: Intel Core Duo, Intel Core i7, Supercomputers, types of supercomputers.

References:

1. William Stallings, Computer Organization and Architecture – Designing for Performance, (11e), Pearson Prentice Hall, 2019.

2. Kai Hwang and Faye A. Briggs, Computer Architecture and Parallel Processing, TMH Private Ltd., 2012.

3. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, (6e), Morgan Kaufmann, 2019.

4. John Paul Shen and Mikko H. Lipasti,, Modern Processor Design: Fundamentals of Superscalar Processors, Tata McGraw-Hill, 2013.

CSE 4453: HUMAN COMPUTER INTERFACE [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Compare the features of humans and computers in reasoning and problem solving.
2. Analyze typical human–computer interaction (HCI) models.
3. Apply interactive processes and universal principles to design HCI systems.
4. Make use of HCI design principles, standards and guidelines.
5. Analyze and identify user models, socio-organizational issues, and stakeholder requirements of HCI systems.

The human: Introduction, Frameworks and HCI, Industrial interfaces, Interaction styles, Navigation in 3D and 2D, Elements of the WIMP interface, The context of the interaction, Half the picture, Experience, engagement and fun. Paradigms, Interaction design basics, HCI in the software process, Design rules, Universal design, Implementation support, Design Focus, Evaluation techniques, User support, Cognitive models: Goal and task hierarchies, GOMS saves money, Linguistic models, The challenge of display-based systems, Physical and device models, Cognitive architectures, Socio-organizational issues and stakeholder requirements: Communication and collaboration models: Introduction, Face-to-face communication, Task analysis

References:

1. Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, *Human–Computer Interaction*, (3e), Edition Pearson, 2014
2. David Benyon, Designing User Experience: A guide to HCI, UX and interaction design, (4e), Pearson 2019.
3. Donald A. Norman, *The design of everyday things*, (2e), Currency and Doubbleday, 2012
4. Rogers Sharp Preece, *Interaction Design:Beyond Human Computer Interaction*, (2e), Wiley 2012
5. Guy A. Boy, *The Handbook of Human Machine Interaction*, Ashgate publishing Ltd, 2011

CSE 4454: INFORMATION RETRIEVAL [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Demonstrate the various models of information retrieval.
2. Formulate queries for efficient retrieval.

3. Identify appropriate storage structures for efficient retrieval.
4. Analyze performance of retrieval system.
5. Assess various search strategies.

Introduction to Information Retrieval and its systems, Information Retrieval Strategies, Boolean Retrieval and Postings Lists, Information Retrieval Problem, Document Delineation and character sequence decoding, Dictionaries and tolerant retrieval and search strategies, Index construction and Index compression, types of indexes, parametric and zone indexes, vector space model, evaluation in information retrieval, relevance assessment, probabilistic information retrieval, Binary independence model, Text classification – classification problems, classification models, clustering in information retrieval, clustering models, Evaluation in clustering, XML retrieval and its strategies, challenges and evaluation of XML retrieval, web crawling and link analysis.

References:

1. Christopher D Manning, Prabhakar Raghavan and Hinrich Schutze, *Introduction to Information Retrieval*, Cambridge University Press, 2008

2. Stefan Buttcher, Charles L.A. Clarke and Gordon V. Cormack., *Information Retrieval – Implementing and Evaluating search engine*s, (6e), MIT Press, 2011

3. Baeza Yates and Ribeiro Neto., *Modern Information Retrieval,* (2e), Addison Wesley, 2010

4. Soumen Charabarti, *Mining the Web*, Morgan-Kaufmann, 2003

5. David A Grossman, OphitFrieder, *Information Retrieval – Algorithms and Heurisitics*, (2e), Springer, 2004

CSE 4455: INFORMATION SECURITY [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Identify issues of privacy, authenticity, security of information, and cyber crimes
2. Recognize different system security threats and countermeasures
3. Analyze various solutions available for secure multilevel database design
4. Identify network security threats, and determine efforts to counter them
5. Apply proper defense mechanisms for various security issues involved in network communication.

What is security? CNSS Security Model, Components of an Information System, Approaches to Information Security Implementation, The Systems Development Life Cycle, Malicious Software Types, Vulnerability Exploit, Social Engineering, System Corruption, Attack Agent, Information Theft, Stealthing, Counter measures, Distributed

Denial of Service Attacks, Intrusion Detection, Need for Firewalls, Database Security, Network Access Control, Extensible Authentication Protocol, IEEE 802.1X Port Based Network Access Control, Cloud security risks and countermeasures, Transport Layer Security, HTTPS, Email threats, Pretty Good Privacy (PGP). IP Security Overview, Policy, Encapsulation, Ways of executing cybercrimes

References:

1. William Stallings, Cryptography and Network Security: Principles and Practice, (7e), Prentice Hall, 2017
2. Michael E. Whitman and Herbert J. Mattord, Principles of Information Security, (4e), Cengage Learning India Publication, 2011.
3. Charles P Pfleeger and Shari Lawrence Pfleeger, Security in Computing, (4e), PHI, 2009
4. Joseph Migga Kizza, A Guide to Computer Network Security, Springer International edition, 2009
5. Atul Kahate, Cryptography and Network Security, Tata McGraw-Hill Publishing, 2008
6. Bruce Schneier, Applied Cryptography-Protocols, Algorithms, and source code in C, (2e), John Wiley & Sons, Inc., 2013

CSE 4456: IOS APPLICATION DEVELOPMENT
[3 0 0 3]

Course Outcomes:

After studying this course, students will be able to:

1. Apply the basic tools and techniques to develop an iOS application using Swift 4.
2. Analyze the fundamental concepts of application development for iOS with the Swift programming language.
3. Design the user interface (UI) and users' interaction for iOS application.
4. Develop a functional iOS application using Swift 4.

Introduction to plyaground; Setting the Apple device for development; Introduction to Swift and Playgrounds; Constants, Variables, and Data Types; Operators; Control Flow; Debugging; Interface Builder Basics; Guided Project - Light; Solving playground, Strings, Functions, Structures, Classes and Inheritance, Collections, Loops, Basics of UIKit, Displaying data, Controls in Actions, Auto Layout and Stack Views, Guided Project - Apple Pie, Solving playground, Optional, Type Casting, Guard, Scope, Enumerations, Segues and Navigation Controllers, Tab Bar Controllers, View Controller Life Cycle, Guided Project - Quiz, Solving playground, Protocols, App Anatomy and Life Cycle, Model View Controller, Scroll Views, Table Views, Intermediate Table Views, Saving Data, System View Controllers, Complex Input Screen, Guided Project - List, Solving playground, Closures, Extensions, Animations, WWW HTTP and URL Session, WWW JSON Serialization, WWW Concurrency, Guided Project -

Restaurant, Solving playground, App Personality, The Design Cycle, Project Planning, App finalization, Unit testing, UI Testing.

References:

1. Mathias M., Gallagher J., Swift Programming: The Big Nerd Ranch Guide (2e), Big NerdRanch Guides, 2016.
2. Nahavandipoor V., iOS 11 Swift Programming Cookbook, O'Reilly Media, 2017.
3. Yamacli S, Beginner's Guide to iOS 11 App Development Using Swift 4: Xcode, Swift andApp Design Fundamentals (1e), USA: CreateSpace Independent Publishing Platform, 2017.
4. Hegarty P, Stanford Engineering CS193p: Developing iOS 10 Apps with Swift, StanfordSchool of Engineering, Stanford University, Melbourne, CA, 2017

CSE 4457: KNOWLEDGE REPRESENTATION AND ONTOLOGY [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1 Analyse the reasons behind foundations and building of semantic web.
2 Model and design real world example in Semantic web.
3 Outline the analysis and design of an ontology through its development and validation.
4 Summarize the different examples and applications of semantic web
5 Apply knowledge representations in different real world applications

From Traditional Web to Semantic Web, Search engines in both traditional and Semantic Web, The Building block of Semantic Web: RDF, RDFS, Taxonomy and Ontology, Validating OWL Ontology, Semantic Web – Real world examples and applications, Knowledge Representation.

References:

1. Stuart Russell and Peter Norvig, Artificial Intelligence- A Modern Approach (3e), Prentice Hall Series, 2010.
2. Liyang Yu, Introduction to Semantic Web and Semantic Web Services, Chapman and Hall/CRC Publications, 2007.
3. Peter Mika, Social Networks and semantic web, Springer, 2007

CSE 4458: MACHINE TRANSLATION [3 0 0 3]

Course outcomes:

After studying this course, students will be able:

1. Outline the basics of Machine Translation
2. Summarize the Word Embedding Schemes
3. Compare the approaches for Machine Translation and draw conclusions
4. Analyze and choose the appropriate model for Machine Translation
5. Solve simple Machine Translation problems

Introduction, Neural Network Basics, Probability, language models, Evaluation of machine translation, Word based translation model, Phrase based Translation model , Neural machine translation, Neural Language models and Translation models, Word embedding, Decoding, Advanced neural MT architectures, Learning from multilingual data, Phrase-based and syntax-based statistical MT, Syntax, Advanced Decoding Techniques, Linguistic Structure, Open challenges.

References:

1. Phillip Koehn, *Neural Machine Translation*, Cambridge University Press, 2020
2. Phillip Koehn, *Statistical Machine Translation*, Cambridge University Press, 2009
3. Emily M. Bender, *Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax*, Morgan & Claypool, 2013
4. Ian Goodfellow, YoshuaBengio and, Aaron Courville, *Deep Learning, The MIT Press*, 2016
5. Dan Jurafsky and James H. Martin, *Speech and Language Processing*, 2nd Edition, Peasrson Education Inc.

CSE 4459: MULTIMEDIA RETRIEVAL [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the basic concepts of multimedia and multimedia retrieval system
2. Analyze the important mathematical techniques in pattern recognition
3. Understand text indexing and retrieval
4. Classify objects in images and detect features
5. Describe video indexing and retrieval

Introduction, Characteristics of Media Data, Metadata of Multimedia Objects, Schematic Overview of MIRS, Quality of an MIRS, Role of IRS, Modeling, Unsupervised Learning and Clustering, Dimension Reduction, Query formulation and matching, Boolean Model, Models for Ranked Retrieval, Term Weighting, Image Features and Feature Extraction, Object Recognition, Generative Probabilistic Models, Combining Visual and Textual

Information, Speech Recognition and Spoken Document Retrieval, Robust Speech Recognition and Retrieval, Audio Segmentation, Cross-media mining, A Case Study on Broadcast News Video, Semantic Pathfinder, Indexing Results on 32 Semantic Concepts, Stroke Recognition Using Hidden Markov Models, Processing the Audio Signal, Detection of Excited Speech using (D)BNs, Analysing the Image Stream, Highlight Detection using DBNs, Superimposed Text, Integrated Querying.

References:

1. Blanken, Henk M., Arjen P. de Vries, Henk Ernst Blok, and Ling Feng, eds. Multimedia retrieval. Springer Science & Business Media, 2007.
2. Feng, David, Wan-Chi Siu, and Hong Jiang Zhang, eds. Multimedia information retrieval and management: Technological fundamentals and applications. Springer Science & Business Media, 2013.
3. Rüger, Stefan. Multimedia information retrieval. Synthesis Lectures on Information Concepts, Retrieval, and Services, 2009.

CSE 4460: MULTIMEDIA TECHNOLOGIES [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the representation of multimedia information and digitization principles.
2. Apply the various text and image compression techniques for data transmission
3. Compare the various audio and video compression techniques and synchronization methods
4. Identify the synchronization methods that could be applied in a distributed environment
5. Analyze the various methods used for multimedia data transmission

Introduction to Multimedia, Graphics and Image Data Representations, Color Models in Images, and Video, Fundamental Concepts in Video, Basics of Digital Audio, Lossless and Lossy Compression Algorithms, Image Compression -JPEG Standard, Basic Audio Compression Techniques, MPEG Audio Compression, Introduction to Video Compression, H.261, MPEG-1, MPEG-4, MPEG 7, Modern Video Coding Standards H.266, Multimedia synchronization and collaboration**,** Synchronization Reference model, Synchronization in a Distributed Environment, Synchronization Specification Methods, Collaboration dimensions, Group Communication Architectures, Session Management, Quality of service for Multimedia Transmission and Interaction, Internet Multimedia Content Distribution, Cloud Computing for Multimedia Services,

References:

1. Li, Ze-Nian, Drew, Mark S., Liu, Jiangchuan,, *Fundamentals of Multimedia*, (3e), Springer 2021.
2. Ralf Steinmetz and Klara Nahrstedt, *Multimedia: Computing, Communications and Applications*, Pearson Education India,2012
3. Ralf Steinmetz and Klara Nahrstedt, *Multimedia Fundamentals- Volume 01 Media coding and content processing*, Pearson Publication, 2008
4. Fred Halsall, *Multimedia Communications- Applications, Networks, Protocols and Standards*, (1e), Pearson Education India,2002
5. K R Rao, Zoran S Bojkovic and Dragorad A Milovanovic, *Introduction to Multimedia Communications*, Wiley Publications, 2009

CSE 4461: PATTERN ANOMALY AND DETECTION [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline anomalies in various domains
2. Summarize the approaches for anomaly detection
3. Apply distance, cluster and model-based methods for anomaly detection
4. Make use of ensemble methods for anomaly detection
5. Analyze the various algorithms to detect anomalies in time series data

Introduction: What is an Anomaly?, Cybersecurity, Finance, Healthcare, Défense and Internal Security, Consumer Home Safety, Manufacturing and Industry, Anomalies, Outliers in One-Dimensional Data, Outliers in Multidimensional Data, Anomaly Detection Approaches, Evaluation Criteria. Distance and cluster based anomaly detection: Introduction, Similarity Measures, Distance-Based Approaches, Identifying clusters, Anomaly Detection using Clusters. Model-based anomaly detection approaches: Models of Relationships between Variables, Distribution Models, Models of Time-Varying Processes, Anomaly Detection in Time Series, Learning Algorithms used to Derive Models from Data. Distance and density based approaches: Distance from the Rest of the Data, Local Correlation Integral Algorithm, Nearest Neighbor Approach, Density Based Approaches. Ensemble methods: Independent Ensemble Methods, Sequential Applications of Algorithms, Ensemble Anomaly Detection with Adaptive Sampling, Weighted Adaptive Sampling. Algorithms for time series data: Problem Definition, Identification of Anomalous Time Series, Abnormal Subsequence Detection, Outlier Detection Based on Multiple Measures, Online Anomaly Detection for Time Series.

References:

1. *Mehrotra, Kishan G., Chilukuri K. Mohan, and HuaMing Huang.* Anomaly detection principles and algorithms. *New York, NY, USA:: Springer International Publishing, 2017.*
2. *Bhattacharyya, Dhruba Kumar, and Jugal Kumar Kalita.* Network anomaly detection: A machine learning perspective. *Crc Press, 2013.*
3. *Dunning, Ted, and Ellen Friedman.* Practical machine learning: a new look at anomaly detection. *" O'Reilly Media, Inc.", 2014.*
4. *Alla, Sridhar, and Suman Kalyan Adari.* Beginning Anomaly Detection Using Python-Based Deep Learning. *Apress, 2019.*
5. *SURI, NNR MURTY RANGA, M. Narasimha Murty, and G. Athithan.* Outlier Detection: Techniques and Applications. *Springer Nature, 2019.*

CSE 4462: PERVASIVE COMPUTING [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Discover the characteristics of pervasive computing applications including the major system components and architectures of the systems
2. Illustrate the role of wireless protocols in shaping the future Internet.
3. Develop pervasive web application architectures
4. Analyze the challenges and opportunities in pervasive computing.
5. Develop an aptitude to propose solutions for problems related to pervasive computing systems.

Introduction to Pervasive Computing: Basics, Characteristics- interaction transparency, context awareness, autonomy experience capture, Pervasive computing infrastructure, Architecture for pervasive computing, Device Technologies, Human-Machine Interfaces, Biometrics, Voice Technologies, Basics of Speech Recognition, Privacy and Security, Energy Constraints, Smart Devices and Services, Sensor Networks, WWW architecture, Protocols, Components of the WAP architecture, WAP infrastructure, WAP security issues, Wireless Mark-up Language, Pervasive Networks, Scalability & Availability, Development of pervasive computing Web Applications, Pervasive Application Architecture, Context aware Computing, Pervasive Computing case studies - Computer-Based Navigation Systems, Smart Homes, Autonomous Vehicles

References:

1. Jochen Burkhardt, *Pervasive Computing: Technology and Architecture of Mobile Internet Applications*, Pearson Education, 2002
2. Stefen Poslad, *Ubiquitous Computing: Smart Devices, Environments and Interactions,* (2e), Wiley, 2010
3. Natalia Silvis-Cividjian, *Pervasive Computing- Engineering Smart Systems*, © Springer International Publishing AG 2017
4. Laurence T. Yang, *Handbook On Mobile And Ubiquitous Computing Status And Perspective*, CRC Press, 2012
5. John Krumm, *Ubiquitous computing fundamentals*, CRC Press, 2016
6. Guruduth S. Banavar, Norman H. Cohen, and Chandra Narayanaswami, *Pervasive Computing: An Application - Based Approach*, Wiley Interscience, 2012
7. A. Genco, S. Sorce, *Pervasive Systems and Ubiquitous Computing*, WIT Press, 2012
8. Minyi Guo, Jingyu Zhou, Feilong Tang, Yao Shen, *Pervasive Computing-Concepts, Technologies and Applications*, © 2017 by Taylor & Francis Group, LLC
9. Deepak Gupta, Aditya Khamparia, *Fog, Edge, and Pervasive Computing in Intelligent IoT Driven Applications,* Copyright © 2021 by The Institute of Electrical and Electronics Engineers, Inc, Hohn Wiley, 2017

CSE 4463: QUANTUM COMPUTING
[3 0 0 3]

Course outcomes:

After studying this course, students will be able to:
1. Analyse quantum model of computation and the basic principles of quantum mechanics.
2. Design and analyse quantum algorithms
3. Demonstrate various quantum protocols.
4. Solve problems using Qiskit tool on quantum computers.
5. Categorie various applications of quantum computing.

Introduction, Linear algebra, Quantum bits, Multiple Bits, Bases and Linear Independence, Linear operators and matrices, The Pauli Matrices, Inner Products, Eigen vectors and Eigen values, Adjoints and Hermitian operators, Tensor Products. Four Postulates of quantum mechanics. Quantum Measurement, Quantum Computation, EPR Paradox, Quantum Gates, Single qubit gates, Multiple qubit gates, Quantum Circuits, Single qubit operations, Controlled operations, Universal quantum gates. No Cloning Theorem, Quantum Key Distribution, Dense coding and Quantum Teleportation. The Quantum Fourier transform (QFT), Applications of QFT: factoring, period finding, Implementation of QFT. The Simple Quantum search algorithms, Deutsch Algorithm, Deutsch-Jozsa Algorithm, Bernstein-Vazirani Algorithm, Simon's Algorithm, Grover's Search Algorithm, Shor's Factorization algorithm, Introduction to Qiskit tool. Quantum codes.

Reference Books:

1. Michael A Nielsen, and Isaac L. Chuang, *"Quantum Computation & Quantum Information"*, (10e), Cambridge University Press, 2011.
2. M. Nakahara and T Ohmi, "*Quantum Computing From Linear algebra to Physical Realizations*" CRC press 2008.
3. Jonathan A. Jones and Dieter Jaksch, "*Quantum Information, Computation and Communication*", Cambridge University Press, 2012.
4. F. Benatti, M. Fannes, R. Floreanini, and D. Petritis, *"Quantum Information, Computation and Cryptography",* Springer, 2010.
5. Mika Hirvensalo, "*Quantum Computing",* (2e), Springer-Verlag New York, 2004.
6. Jozef Gruska, "*Quantum Computing"*, McGraw Hill, 1999.
7. Phillip Kaye, Raymond Laflamme and Michele Mosca, "*An Introduction to Quantum Computing*", Qxford University Press, 2006.

CSE 4464: SOCIAL NETWORK ANALYSIS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Summarize the basic concepts of network structure and representation of Social Network Analysis.
2. Analyze the Social Network structure and visualize them in the form of layouts
3. Apply the Social Network Concepts in solving problems related to social, personal, business and international levels
4. Evaluate the algorithms for discovering communities in Social Networks.
5. Interpret the models for social influence analysis.

Introduction to Social Web, Social Networks Vs Link analysis, the power of Informal networks Vs Social Networks,Nodes, Edges and Network Measures, Describing Nodes and Edges, Describing Networks, Layouts, Visualizing network features, The role of Tie strength, Measuring Tie strength and its network structures, network propagation, Link prediction, entity resolution, Case study, Introduction to community discovery, communities in context, quality functions, The Kernighan-Lin algorithm, Agglomerative algorithms, spectral algorithms, multi-level graph partitioning, Markov clustering, Other approaches, Introduction to social influence, Influence related statistics, social similarity and influence, Homophily, Existential Test for social influence, Influence and actions, Influence and interactions, influence maximization in viral marketing, Triads, directed triads, analysing triads in Real networks, Real data, detecting cliques.

1.

References:

1. Jennifer Golbeck., *Analysing the Social Web*, Morgan Kaufmann publications, 2013

2. Charu C. Aggarwal, *Social Network Data Analytics*, Springer publications, 2011.
3. Maksim Tsvetovat and Alexander Kouznetsov, *Social Network Analysis for Startups*, O'Reilly Media, 2011.
4. John Scott, *Social Network Analysis,* (3e), Sage publications limited, 2013
5. Jay Goldman, *Facebook Cookbook,* O'Reilly, 2009
6. Shamanth Kumar, Fred Morstatter, Huan Liu, *Twitter Data Analytics*, Springer publications, 2013

CSE 4465: SOFTWARE ARCHITECTURE [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the techniques and methods of Software Architecture.
2. Analyze general purpose component and middleware technologies.
3. Identify the changes in technologies and practices that may affect the next generation of Business Information System.
4. Compare the software architecture evaluation techniques.
5. Construct service-oriented architecture and technologies.

Understanding Software Architecture, Definitions of Software Architecture, Architectures and Technologies, Introducing the Case study, Requirements Overview, Project Context, Business Goals, Software Quality Attributes, Performance, Scalability, Modifiability, Security, Availability, Integration, Other Quality Attributes, Design Trade-offs, Middleware Architectures and Technologies, Application Servers, Software Architecture Process, Documenting Software Architecture, Case Study, Design and The Challenges of complexity, Aspect Oriented Architectures, Model Driven Architecture, Service Oriented Architectures and Technologies, The Semantic Web.

References:

1. Gorton Ian, *Essential Software Architecture, (2e)*, Springer International Architecture, 2011
2. Bass Len, Clements Paul, Kazman, Rick, *Software Architecture in Practice,* (2e), Pearson, 2003
3. Bosch Jan, *Design and Use of Software Architecture,* Addison Wesley, 2000
4. Rozanski Nick & Woods Eoin, *Software Systems Architecture*, Addison Wesley, 2005

5. Mark Richards, Neal Ford, *Fundamentals of Software Architecture An Engineering approach*, O'Reilly Media, Inc., 2020.

CSE 4466: SOFTWARE DEFINED NETWORKS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the evolution of Software Defined Networks
2. Analyze the functions and components of the SDN architecture.
3. Create an SDN network consisting of SDN switches and a controller using OpenDaylight.
4. Identify a network policy using SDN for specific applications.
5. Illustrate the network function virtualization and network security.

History and Evolution of SDN, Needs of SDN, Traditional Switch Architecture, Control and data plane basics, Control and Data Plane Separation: How SDN Works – Centralized and Distributed Control and Date Planes, IP and MPLS, Creating the IP Underlay, Creating the MPLS Overlay, Route Servers, Introducing OpenFlow: OpenFlow Architecture, OpenFlow Versions, OpenFlow Messages, Symmetric messages, Asynchronous Messages, Implementing the OpenFlow Switch: OpenFlow Reference Switch, OpenFlow Controllers: Setting up an environment: Understanding the OpenFlow Laboratory OpenDayLight and designing SDN Applications, Network Function Virtualization: Introduction, Virtualization and data plane I/O, NFV at ETSI, SDN Security: Security Requirements - SDN Security - NFV Security - ETSI Security Perspective - IoT Security - The Patching Vulnerability - IoT Security and Privacy, An IoT Security Framework

References:

1. Thomas D. Nadeau, Ken Gray, “SDN: Software Defined Networks, An Authoritative Review of Network Programmability Technologies, O'Reilly MediaISBN: 978-1-4493-4230-2, ISBN 10:1-4493-4230-2, (1e), 2013.
2. Saimak Azodolmolky, “Software Defined Networking with OpenFlow, ”Packt Publishing, (2e), 2017.
3. Paul Goransson and Chuck Black, “Software Defined Networks: A Comprehensive Approach, (1e), Morgan Kaufmann, 2014.
4. William Stallings, “Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud”, (1e), Addison-Wesley, 2015
5. Rajesh Kumar Sundararajan, “Software Defined Networking-A definitive guide, Kindle book, (1e), 2013.

CSE 4467: SOFTWARE TESTING AND ANALYSIS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Explain various test processes and continuous quality improvement.
2. Apply functional testing criteria to improve the quality of software.
3. Design test cases using structural testing techniques.
4. Evaluate the adequacy of test suite using mutation testing.
5. Plan and execute a testing strategy for a software project

Introduction, Software Quality, Behaviour and Correctness, Correctness versus Reliability, Testing and Debugging, V-Model, Boundary Value Analysis, Robustness Testing, Worst Case Testing, Special Value Testing, Random Testing, Equivalence Class Testing, Decision Tables, Test Cases based on Decision Tables, Test adequacy basics, Adequacy criteria based on control flow, data flow concepts, Adequacy criteria based on data flow, Test Assessment using Mutation, Regression Test Process, Selecting Regression Tests, Test Design, Using JUnit or NUnit, Stubs and Mocks, Integration errors, Dependence, OO Versus Non-OO programs, Integration Hierarchy, Automated test generation techniques.

References:

1. Aditya P Mathur, *Foundations of Software Testing*, (2e), Pearson Education, 2008
2. Paul C. Jorgensen, *Software Testing A Craftsman's Approach*, (3e), 2013
3. Mauro Pezze, Michal Young, *Software Testing and Analysis: Process, Principles and Techniques*, Wiley, 2008
4. Gopalaswamy Ramesh, Srinivasan Desikan, *Software testing Principles and Practices*, (2e), Pearson, 2007

CSE 4468: STORAGE DEVICE AND TECHNOLOGY [3 0 0 3]

Course outcomes:

After studying this course, students will be able to

1. Evaluate storage architectures and key data center.
2. Explain physical and logical components of a storage infrastructure.
3. Outline storage options and protocols.
4. Develop business continuity solutions.
5. Demonstrate the backup and replications, along with archive for managing the data

Evolution of Storage Technology and Architecture, Data Center Infrastructure, Virtualization and Cloud computing. Components of a Storage System Environment, Disk drive Performance, Logical Components of the Host, Direct-Attached storage, Disk performance, command queuing, flash drives. RAID Techniques, RAID Levels, Hot Spares. Types of Intelligent Storage Systems, Components of FC SAN, Switched Fabric Ports, World Wide Names, Zoning, Switched Fabric Login Types, Components of NAS, Object Storage and Retrieval in OSD, Benefits, Content addressed storage, Information Availability, BC Terminology, Backup Purpose and Considerations, Recovery Considerations, Local Replication Technologies, Remote Replication Technologies, Three-Site Replication.

References:

1. G.Somasundaram, Alok Shrivastava, *Information Storage and Management-Storing, Managing, and Protecting Digital Information in classic, virtualized and cloud environments, (2e),* EMC Education Services, John Wiley & Sons Inc., 2012

2. Marc Farley, *Storage Networking Fundamentals, (1e),* CISCO Systems, 2004

3. Robert Spalding, *Storage Networks: The Complete Reference,* Tata Mcgraw Hill, 2003

4. Marc Farley Osborne, *Building Storage Networks, (2e),* Tata McGraw Hill, 2001

CSE 4469: UML AND DESIGN PATTERNS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Identify the relevant UML diagrams required for software design
2. Translate the software requirements into software design using UML diagrams
3. Extend the knowledge of software design using design patterns
4. Justify the required design pattern for solving a given problem
5. Propose the best design pattern for software development

The UML, Use Case, Class and Object, Sequence, Collaboration, State chart, Activity, Component and Deployment Diagrams. Other Notation and information, Associations, Links, Compositions, Use Case Diagram, Actor, Use case, Sequence Diagram-

Interactions, Class Roles, Collaboration Diagram- Collaborations, Association Roles, State chart Diagram- States, Transitions, Activity Diagram- Case study - Swimlanes, Action State, Component Diagram- Components, Deployment Diagram- Nodes, Communication Relationships. Introduction of Design Pattern, Creational Pattern, Structural Pattern, Conclusion – What to expect from the design pattern, The pattern community.

References:

1. Sinan Si Alhir, *UML in a Nutshell*, O'Reilly, 2004.
2. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Pearson Education, 2007
3. Booch Grady, Rambugh James, Jacobson Ivar, *The Unified Modeling Language User Guide*, Second Edition, Pearson Education, 2010.
4. G Lasater, *Design patterns*, Wordware publishing Inc, 2007.
5. Jim Conallen, *Building Web Applications with UML*, 2e, Addison-Wesley Professional, 2002.

CSE 4470: WIRELESS NETWORKS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the concepts of wired and wireless communication systems
2. Examine the working of various cellular networks.
3. Illustrate the various protocols of wireless networks.
4. Analyze the routing protocols for multi-hop wireless networks.
5. Evaluate the importance of IP multimedia subsystems in wireless networks

Introduction to Wireless Networks: Evolution, Challenges, Introduction to Wireless Communication, Electromagnetic spectrum, Spectrum regulation, Wireless propagation, Modulation techniques, Multiple access for wireless systems, Cellular concept, Wireless services Ubiquitous Connectivity, Types of Wireless Networks, Analog Cellular Systems, AMPS, DAMPS, GSM, 3G Spectrum allocation, CDMA, WCDMA, 4G and Beyond, OFDM, Fixed Wireless Networks. IEEE 802.16, WLAN, applications, topology, requirements, Physical and MAC layers, IEEE 802.11a,b and g, Ad Hoc Networks: topology, Ad Hoc routing, VANETs, IMS architecture, IMS Call Flow, services within IMS, The 5G Internet, 5G Mobile Networks

References:

1. R Nicopolitidis et al, *Wireless networks*, Wiley, first, 2011

2. Jeffrey Bannister, Paul Mather and Sebastian Coope *"Convergence technologies for 3G networks, IP, UMTS, EGPRS and ATM"*, John Wiley & Sons Publications, 2004
3. Jonathan Rodriguez, Fundamentals of 5G Mobile Networks, Wiley, 2015
4. Kaveh Pahlavan, Prashant Krishnamurthy, "*Principles of Wireless Networks*", Prentice Hall, 2011
5. William Stallings ,"*Wireless Communication and Networking*", PHI, 2014

Open Electives

CSE 4311: ESSENTIALS OF INDUSTRIAL COMPUTING [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Apply object-oriented concepts.
2. Demonstrate programming concepts.
3. Analyze different algorithms.
4. Design different software development models.
5. Outline the security and web application issues.

Introduction, Programming techniques, Introduction to object-oriented concepts, Object-oriented concepts – Inheritance and Polymorphism, Object oriented design methodology. Introduction to Analysis of algorithms, Code tuning techniques, Analysis of algorithms, Analysis of some well-known algorithms, Intractable problems. Evolution of software, Software development models, Requirement analysis and design, Software construction, Software testing, Software quality. Computer Networks, Introduction to Web Technologies, Internetworking concepts, Architecture and Protocol, The World Wide Web, Web applications, Security in applications, Issues in Web Based Applications.

References:

1. Foundation Program Team, Foundation Program – Vol. 2, (3e), Education and Research, Infosys Technologies, 2012.
2. Foundation Program Team, Foundation Program – Vol. 3, (3e), Education and Research, Infosys Technologies, 2012.
3. Comer Douglas E, Computer Networks and Internets, (6e), Pearson, 2014.

CSE 4312: ESSENTIALS OF IT [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the fundamentals of computer architecture and operating systems.
2. Summarize the memory management and process management in operating system.
3. Illustrate the working of file systems and managing I/O devices.
4. Create database using RDBMS concepts.
5. Make use of SQL commands in database projects.

Fundamentals of Computer Architecture, Organization of a Simple Computer, Execution of Instructions, Input/Output Devices, Measurement of CPU Performance, Operating System Concepts, Memory Management, Process Management, Basic Concept of Multiprogramming, Multitasking and Multiprocessing, Interprocess Communication, File Management and Device Management, Disk Scheduling, Introduction to DBMS and Entity Relationship Modelling, Converting ER diagram to Schema, Introduce Functional Dependency and Basics of Normalization, Introduce three Normal Forms, SQL, Data Definition Language (DDL) statements, Data Manipulation Language (DML) statements and Data Control Language (DCL).

References:

1. Foundation Program Team, *Foundation Program – Vol. 1* (3e), Education and Research, Infosys Technologies, 2012
2. Foundation Program Team, *Foundation Program – Vol. 2* (3e), Education and Research, Infosys Technologies, 2012
3. Abraham Silberschatz, Peter B. Galvin and Greg Gagne, *Operating System Concepts* (9e), John Wiley & Sons, 2018
4. Milenkovic Milan., *Operating Systems Concepts and Design* (2e), McGraw Hill Education, 2001
5. Henry F Korth, Abraham Silberschatz and Sudarshan, *Database system concepts* (7e), McGraw-Hill Education, 2019.

CSE 4313: LINUX PROGRAMMING [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Implement shell scripts to perform various tasks in Linux System.
2. Build various patterns using awk and sed
3. Demonstrate the usage of debugger in Linux environment.

4. Create the source code with version control systems.
5. Apply Linux platform for document creation using Latex tools.

Understanding what Linux is, About shells and terminal Windows, Choosing shell, running commands, recalling Commands, using command History, connecting and expanding commands, using shell variables, creating your shell environment, getting information about commands, using basic filesystem commands, understanding file permissions and ownership, patterns, metacharacters, understanding system administration, editing files with vim and vi, The sed and awk programming, shell programming, programming in Linux, Makefiles, debugging, writing and using libraries, source code management using git version control system, using Latex in preparing scientific documentations & in preparation of presentation slides

References:

1. Christopher Negus, *Linux Bible*, (8e), John Wiley & Sons Publication, 2012
2. Ellen Siever, Stephen Figgins, Robert Love, and Arnold Robbins, *Linux in a Nutshell*, (6e), O'Reilly Media Publication,2009
3. Mark Mitchell, Jeffrey Oldham, and Alex Samuel, *"Advanced Linux Programming"*, New Riders Publishing, 2001
4. Leslie Lampert, *Latex - A document preparation system,* (2e), Addison-Wesley,1994.
5. Sumitabha Das, *UNIX Concepts and Applications,* (4e), Tata McGraw Hill Publications,2011.

CSE 4314 : PRINCIPLES OF DATABASE SYSTEMS [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Outline the basic concepts of database and relational model
2. Apply structured query language for data retrieval
3. Utilize E-R models to design databases
4. Analyze and apply the normalization technique to decompose given relational schema into effective schemas
5. Explain the method of indexing, hashing and organization of files and apply the different transaction properties for serializability.

Database-System Applications, Database Languages, Relational Databases, Database Design, Data Storage and Querying, Database Architecture, Database Schemas, Keys, SQL Data Definition, SQL Data Types and Schemas, Integrity Constraints, Set Operations, Aggregate Functions, Overview of SQL Query Language, Basic Structure of SQL Queries, Join Expressions, Overview of the Design Process, The Entity-Relationship Model, Extended E-R Features, Reduction to Relational Schemas, Features of Good Relational Design, Atomic Domains and Normalization, File concepts, Indices Concept, Transaction Isolation Serializability, Transaction Isolation and Atomicity, Transaction Isolation Levels.

References:

1. Silberschatz, Korth, Sudarshan, *Database System Concepts*, (6e), McGrawHill, New York, 2011.
2. Ramez Elmasri and Shamkant Navathe, Durvasula V L N Somayajulu, Shyam K Gupta, *Fundamentals of Database Systems*, (6e), Pearson Education, United States of America, 2011

CSE 4315: PRINCIPLE OF SOFTWARE ENGINEERING [3 0 0 3]

Course outcomes:

After completing this course, students will be able to:

1. Explain principles of software Engineering
2. Analyze requirements and estimate the cost of software project
3. Identify, design, and develop software process models
4. Build and test DFDs and UML Diagrams
5. Implement, test and analyze software projects.

The Problem Domain, The Software Engineering Challenges, The Software Engineering Approach. Software Process, Desired Characteristics of Software Process, Software Development process models. Software Requirements, Problem Analysis, Requirement Specifications, Functional Specification with Use Cases, Process Planning, Effort Estimation, Project Scheduling and Staffing, Software Configuration Management Plan. Design Principle, Module Level Concepts, Design Notations and Specifications, Structured Design Methodology. OO Analysis and OO Design, OO Concepts, Unified

Modelling Language. Programming Principle and Guidelines, Coding Process, Testing Fundamentals, Black Box Testing, White Box Testing, Testing Process.

References:

1. Pankaj Jalote, *An integrated approach to software engineering (3e)*, Narosa 2005.
2. Rajib Mall , *Fundamentals of Software Engineering (5e)*, PHI learning 2018.
3. Bruce R Maxim, Roger S. Pressman, *Software Engineering A Practioner's Approach (8e)*, McGraw-Hill, 2019.
4. Ian Sommerville, *Software Engineering (10e)*, Pearson, 2015.
5. Hans van Vliet, *Software Engineering: Principles and Practice (3e)*, John Wiley & Sons, 2010.

CSE 4316: PYTHON PROGRAMMING [3 0 0 3]

Course outcomes:

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

1. Apply core data types in Python such as lists, tuples, dictionaries, class and strings.
2. Analyze retrieved data with appropriate Python visualization libraries.
3. Build simple graphical user interfaces in Python.
4. Make use of Python database API to create simple database applications
5. Design web applications to create interactive web pages

Getting started with python scripting, Using the file system, Reading and writing files, Numerical Computing In Python, SciPy package, Classes and object-oriented programming, Data types as objects, Graphical user interfaces, Regular expressions, Network, web, and database programming: Accessing databases in Python, Network programming in Python, Creating a Python web application, Sample project—creating a message wall, Web frameworks creating a model to add database service – using SQLite; Cloud computing: google app engine and web services: What is cloud computing, levels of cloud computing service, what is AappEngine, The sandbox and the App Engine SDK, Choosing an App Engine framework.

References:

1. Hans Peter Langtangen, *Python Scripting for Computational Science*, (3e), Springer Publishers, 2014
2. Naomi R. Ceder, *The Quick Python Book*, (2e), Manning Publications Co., 2010
3. Wesley J. Chun, *Core Python Applications Programming*, (3e), Prentice Hall Publishers, 2012
4. Bill Lubanovic, *Introducing Python - Modern Computing in Simple Packages*, O'Reilly Publication, 2015
5. Allen B. Downey, *Think Python-How to think like a computer scientist*, (2e) O'Reilly Publication, 2015

CSE 4317: WEB PROGRAMMING [3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Design static web pages using HTML5.
2. Create consistent and intuitive UI using CSS code
3. Develop dynamic features in web pages using JavaScript
4. Build client side functionality using jQuery framework.
5. Summarize the knowledge of HTML5, CSS3, and jQuery to build real world applications

Introduction to HTML5 and CSS3, Markup, HTML5 Style, More HTML5 Semantics, HTML5 Forms, HTML5 Audio and Video. Introducing CSS3, CSS3 Gradients and Multiple, CSS3 Transforms and Transitions, Embedded Fonts and Multicolumn Layouts. Introduction to JavaScript, The Grammar of JavaScript, Adding Logic and Control to Your Programs. Introduction to jQuery, Action/Reaction: Making Pages Come Alive with Events, Animations and Effects, Common jQuery Tasks, Enhancing Web Forms.

References:

1. Alexis Goldstein, Louis Lazaris, Estelle Weyl, *HTML5 and CSS3 for The Real World*, (2e), SitePoint, 2015
2. David Sawyer McFarland, *JavaScript and jQuery The Missing Manual*, (3e), O'Reilly Media, Inc., 2014
3. Matthew MacDonald, *HTML5: The Missing Manual*, (2e), O'Reilly Media, 2013
4. Jon Duckett, Gilles Ruppert, Jack Moore, J*avaScript and JQuery: Interactive Front-End Web Development*, John Wiley & Sons, 2014

5. Ed Tittel, Chris Minnick, *Beginning HTML5 & CSS3 for Dummies*, A Wiley Brand, 2013

CSE 4318: IOS APPLICATION DEVELOPMENT

Course Outcomes:

After studying this course, students will be able to:
1. Apply the basic tools and techniques to develop an iOS application using Swift 4.
2. Analyze the fundamental concepts of application development for iOS with the Swift programming language.
3. Design the user interface (UI) and users' interaction for iOS application.
4. Develop a functional iOS application using Swift 4.

Introduction to plyaground; Setting the Apple device for development; Introduction to Swift and Playgrounds; Constants, Variables, and Data Types; Operators; Control Flow; Debugging; Interface Builder Basics; Guided Project - Light; Solving playground, Strings, Functions, Structures, Classes and Inheritance, Collections, Loops, Basics of UIKit, Displaying data, Controls in Actions, Auto Layout and Stack Views, Guided Project - Apple Pie, Solving playground, Optional, Type Casting, Guard, Scope, Enumerations, Segues and Navigation Controllers, Tab Bar Controllers, View Controller Life Cycle, Guided Project - Quiz, Solving playground, Protocols, App Anatomy and Life Cycle, Model View Controller, Scroll Views, Table Views, Intermediate Table Views, Saving Data, System View Controllers, Complex Input Screen, Guided Project - List, Solving playground, Closures, Extensions, Animations, WWW HTTP and URL Session, WWW JSON Serialization, WWW Concurrency, Guided Project - Restaurant, Solving playground, App Personality, The Design Cycle, Project Planning, App finalization, Unit testing, UI Testing.

References:

1. Mathias M., Gallagher J., *Swift Programming: The Big Nerd Ranch Guide (2e),* Big NerdRanch Guides, 2016.
2. Nahavandipoor V., *iOS 11 Swift Programming Cookbook*, O'Reilly Media, 2017.
3. Yamacli S, *Beginner's Guide to iOS 11 App Development Using Swift 4: Xcode, Swift and App Design Fundamentals (1e),* USA: CreateSpace Independent Publishing Platform, 2017.
4. Hegarty P, *Developing iOS 10 Apps with Swift*, Stanford School of Engineering, Stanford University, Melbourne, CA, 2017

Inter-Institute Elective (IIE)

IIE XXXX: Healthcare IT[3 0 0 3]

Course outcomes:

After studying this course, students will be able to:

1. Choose components of health care delivery and health care systems.
2. Interpret general functions, purposes and benefits of HealthCare Information Technology (HIT) systems in various health care settings.
3. Identify how electronic health records affect patient safety, quality care, efficiency, productivity, and reporting/documentation mechanisms.
4. Outline common components of a HIT system, types of HIT applications and illustrate data flows across HIT systems.
5. Build small software applications using Open APIs for select use cases of HIT.

Healthcare Domain Basics, Introduction and History of Modern Health Care, Delivering Health Care, Financing Health Care , Regulating Health Care , Meaningful Use, Health Management Information Systems Overview, Electronic Health Records, Clinical Decision Support Systems, Patient Monitoring Systems, Medical Imaging Systems, Consumer Health Informatics, Administrative, Billing, and Financial Systems, Working with Health IT Systems, Introduction & Overview: Components of HIT Systems, Understanding Information Exchange in HIT System, Protecting Privacy, Security, and Confidentiality in HIT Systems, HIT System Planning, Acquisition, Installation, &Training: Practices to Support & Pitfalls to Avoid, Potential Issues with Adoption and Installation of a HIT system, HIT and Aspects of Patient-Centered Care, Health IT in the Future, Tools, technology used in Healthcare IT, AI/ML use cases in Healthcare IT, Open API usage to build Healthcare IT applications

References :

HealthIT.gov

1. https://files.healthit.gov/Component_1/Comp1_Component_Guide.docx
2. https://files.healthit.gov/Component_6/Comp6_ComponentGuide.docx
3. https://files.healthit.gov/Component_7/Comp7_ComponentGuide.docx