CENTRAL UNIVERSITY OF PUNJAB
BATHINDA

M. Tech. Computer Science & Technology

Session - 2021-23

Department of Computer Science & Technology

Programme Educational Learning Outcomes

1. To build a rich intellectual potential embedded with inter-disciplinary knowledge, human values and professional ethics among the youth, aspirant of becoming technologists, so that they contribute to society and create a niche for a successful career.
2. To enable students to gain research and development competence to sustain in academia as well as industry.
3. To Produce "Creators of Innovative Technology".

Graduate Attributes:

After the Completion of Graduate Program student will be able:

1. To demonstrate competence in engineering mathematics, engineering fundamentals, and specialized engineering knowledge appropriate to the program.
2. To acquire appropriate knowledge and skills to identify, formulate, analyze, and solve computer engineering problems in order to reach substantiated conclusion.
3. To conduct investigations of problems by appropriate experiments, analysis and interpretation of data and synthesis of information in order to reach valid conclusions.
4. To design solutions for open-ended engineering problems for designing systems, components or processes that meet specified needs of program.
5. To create, select, apply, adapt, and extend appropriate techniques, resources, and modern engineering tools for a range of engineering activities.
6. To work effectively as a member and leader in teams, preferably in a multi-disciplinary setting.
7. To understand the role of engineer with professional and ethical responsibilities in the society for public interest.
8. To analyze social and environmental aspects of engineering activities.
9. To communicate complex engineering concepts within the profession and with society at large.
10. To appropriately incorporate economics and business practices including project, risk, and change management into the practice of engineering and to understand their limitations.
11. To identify and address their own educational needs in a changing world in ways sufficient to maintain their competence and advancements in future.
12. To apply professional ethics, accountability and equity.

Program Outcome

After the completion of degree program student will be able:

1. To apply mathematical foundations, algorithmic principles, and computer science theory in the modelling and design of computer based systems.
2. To apply the engineering knowledge in all domains, viz., health care, banking and finance, other professions such as medical, law, etc.
3. To design and conduct experiments as well as to analyze and interpret data.

4. To analyze the problem, subdivide into smaller tasks with well-defined interface for interaction among components, and complete within the specified time frame and financial constraints.
5. To propose original ideas and solutions, culminating into a modern, easy to use tool, by a larger section of the society with longevity.

Course Structure of M.Tech Computer Science & Technology
SEMESTER-I

Course Code	Course Title	Course Type	Credit Hours			
			L	T	P	Cr
CST.506	Advanced Data Structures	Core	4	0	0	**4**
CST.507	Mathematical Foundation of Computer Science	Core	4	0	0	**4**
Elective I						
CST.509	Wireless Sensors Networks	Any one Discipline Elective	4	0	0	**4**
CST.526	Python Programming for Data Sciences					
CST.510	Compiler for HPC					
Elective II						
CST.511	Distributed Database System	Any one Discipline Elective/MOOC	4	0	0	**4**
CST.512	Information Security					
CST.513	Software Testing & Maintenance					
CST.606	Research Methodology and IPR	Compulsory Foundation	4	0	0	**4**
XX.YYY	Opt any one course from the courses offered by the University	IDC	2	0	0	**2**
CST.515	Advanced Data Structures – Lab	Skill Development	0	0	2	**1**
CST.516	Wireless Sensors Networks Lab	Skill Development	0	0	2	**1**
CST.528	Python Programming for Data Sciences Lab					
CST.518	Compiler for HPC Lab					
CST.514	Distributed Database System Lab	Skill Development	0	0	2	**1**
CST.519	Information Security Lab					
CST.520	Software Testing & Maintenance Lab					
Total Credits			**22**	**0**	**6**	**25**

List of IDC for other departments (Semester-I)

Course Code	Course Title	Course Type	Credit Hours			
			L	T	P	Cr
CBS.518	IT Fundamentals	Interdisciplinary courses offered by CST Faculty (For students of other Departments)	2	0	0	**2**
CBS.519	Programming in C					
CST.530	Introduction to Digital Logic					
CST.531	Multimedia and its Applications					
CST.532	Introduction to MatLab					
Total Credits			**2**	**0**	**0**	**2**

Course Structure of M.Tech Computer Science & Technology
SEMESTER-II

Course Code	Course Title	Course Type	Credit Hours			
			L	T	P	Cr
CST.521	Advanced Algorithms	Core	4	0	0	**4**
CST.522	Soft Computing	Core	4	0	0	**4**
Elective III						
CST.523	Computer Vision	Discipline Elective	4	0	0	**4**
CBS.524	Big Data Analysis and Visualization					
CBS.523	Secure Software Design					
CST.524	IOT (Internet of Things)					
Elective IV						
CBS.525	Secure Coding	Discipline Elective	4	0	0	**4**
CST.525	GPU Computing					
CST.529	Blockchain Technology					
CBS.527	Digital Forensics					
CBS.530	Quantum Computing & Cryptography					
CST.508	Machine Learning	Skill Development	4	0	0	**4**
XXX.YYY	Any VAC Course offered by the University	Value Aided either as Theory# or Practical##	2#	0	4##	**2**
CST.527	Soft Computing-Lab	Skill Development	0	0	2	**1**
CST.533	Computer Vision Lab	Skill Development	0	0	2	**1**
CBS.534	Big Data Analysis and Visualization Lab					
CBS.539	Secure Software Design Lab					
CST.534	IOT (Internet of Things) Lab					
CBS.536	Secure Coding Lab	Skill Development	0	0	2	**1**
CST.535	GPU Computing Lab					
CST.536	Blockchain Technology Lab					
CBS.535	Digital Forensics Lab					
CBS.538	Quantum Computing & Cryptography Lab					
CST.517	Machine learning – Lab	Skill Development	0	0	2	**1**
Total Credits			**20/22#**	**0**	**8/12##**	**26**

List of Value Added Courses

Course Code	Course Title	Course Type	Credit Hours			
			L	T	P	Cr
CST. 504	Python Programming*	Value added Course	2	0	0	2
CBS.504	Report Writing using LaTex	Value added Course	2	0	0	2

*** For other departments only**

Course Structure of M.Tech Computer Science & Technology
SEMESTER-III

Course Code	Course Title	Course Type	Credit Hours			
			L	**T**	**P**	**Cr**
CST.551	Optimization Techniques	Any one Discipline Elective/MOOC	4	0	0	**4**
CST.552	Data Warehousing and Data Mining					
CST.553	Intelligent System					
CST.554	Mobile Applications & Services					
CBS.552	Cyber Threat Intelligence	Open Elective/MOOC (Select any one from list)	4	0	0	**4**
CST.556	Cost Management of Engineering Projects					
CBS.553	Cyber Law					
CST.557	Software Metrics					
CST.559	Capstone Lab	Core	0	0	4	**2**
CST.600	Dissertation/ Industrial Project	Core	0	0	20	**10**
Total Credits			**8**	**0**	**24**	**20**

Students going for Industrial Project/Thesis will complete these courses through MOOCs.

L: Lectures T: Tutorial P: Practical Cr: Credits

Course Structure of M.Tech Computer Science & Technology
SEMESTER-IV

Course Code	Course Title	Course Type	Credit Hours			
			L	**T**	**P**	**Cr**
CST.600	Dissertation	Core	0	0	32	**16**
Total Credits			**0**	**0**	**32**	**16**

Mode of Transaction: Lecture, Laboratory based Practical, Seminar, Group discussion, Team teaching, Self-learning.

Evaluation Criteria for Theory Courses/or As per University Pattern

A. Continuous Assessment/Internal Assesment: [25 Marks]
B. Mid Semester Test-1: Based on Subjective Type Test [25 Marks]
C. End Semester Test: Based on Subjective Type Test(70%) and Objective(30%) [50 Marks]

*Every student has to take up one ID courses of 02 credits from other disciplines in semester I of the program and Value Added Course of 2 credits in Semester II.

SEMESTER – I

L	T	P	Cr
4	0	0	4

Course Code: CST.506
Course Title: Advanced Data Structures

Total Hours: 60

Course Objectives:
The objective of this course is to provide the in-depth knowledge of different advance data structures. Students should be able to understand the necessary mathematical abstraction to solve problems. To familiarize students with advanced paradigms and data structure used to solve algorithmic problems.

Course Outcomes
After completion of course, students would be able to:

- Describe various types of data structures and list their strengths and weaknesses.
- Classify non-randomized and randomize algorithms.
- Use data structures for various applications.
- Summarize suitable data structure for computational geometry problems.

UNIT I **14 Hours**

Introduction to Basic Data Structures: Importance and need of good data structures and algorithms.
Dictionaries: Definition, Dictionary Abstract Data Type, Implementation of Dictionaries.
Hashing: Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing.
Activities: Web based simulation

UNIT II **16 Hours**

Skip Lists: Need for Randomizing Data Structures and Algorithms, Search and Update Operations on Skip Lists, Probabilistic Analysis of Skip Lists, Deterministic Skip Lists.
Binary Search Trees, AVL Trees, Red Black Trees, 2-3 Trees, B-Trees, Splay Trees.
Activities: Visual Modelling of data structure

UNIT III **16 Hours**

String Operations, Brute-Force Pattern Matching, The Boyer-Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Standard Tries, Compressed Tries, Suffix Tries, The Huffman Coding Algorithm, The Longest Common Subsequence Problem (LCS), Applying Dynamic Programming to the LCS Problem.
Activities: Web based Training via simulation modelling

UNIT IV **14 Hours**

Computational Geometry: One Dimensional Range Searching, Two Dimensional Range Searching, constructing a Priority Search Tree, Searching a Priority Search Tree, Priority Range Trees, Quad trees, k-D Trees.

Recent Trends in Hashing, Trees, and various computational geometry methods for efficiently solving the new evolving problem.
Activities: Implementation and solution of algorithms, case study of recent trends in algorithms.

Transactional Modes:

- Lecture
- Blended Learning
- Collaborative Learning
- Peer Learning/Teaching
- Online Teaching Tools

Suggested Readings:

1. Cormen, T.H., Leiserson, C. E., Rivest, R.L., and Stein, C. (2015). Introduction to Algorithms. New Delhi: PHI Learning Private Limited.
2. Sridhar, S. (2014). Design and Analysis of Algorithms. New Delhi: Oxford University Press India.
3. Allen Weiss M. (2014). Data Structures and Algorithm Analysis in C++. New Delhi: Pearson Education.
4. Goodrich M.T., Tamassia, R. (2014). Algorithm Design. United States: Wiley.
5. Aho, A.V., Hopcroft, J.E. and Ullman, J.D. (2013). Data Structures and Algorithms. New Delhi: Pearson Education.
6. Horowitz, E., Sahni, S. and Rajasekaran, S. (2008). Fundamentals of Computer Algorithms. New Delhi: Galgotia Publications.
7. Benoit, Anne, Robert, Yves, Vivien and Frederic. (2014). A guide to algorithm design: Paradigms, methods and complexity analysis. London: CRC Press Taylor & Francis group.
8. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.507
Course Title: Mathematical Foundation of Computer Science
Total Hours: 60

Course Objectives:
To make students understand the mathematical fundamentals that is prerequisites for a variety of courses like Data mining, Network protocols, analysis of Web traffic, Computer security, Software engineering, Bioinformatics, Machine learning. To develop the understanding of the mathematical and logical basis to many modern techniques in information technology like machine learning, programming language design, and concurrency.

Course Outcomes
After completion of course, students would be able to:

- Describe the basic notions of discrete and continuous probability.
- Explain the methods of statistical inference, and the role that sampling distributions play in those methods.
- Employ correct and meaningful statistical analyses of simple to moderate complexity problems.
- Categorize the domain specific mathematical models for different analysis.

UNIT I **16 Hours**
Distribution Function: Probability mass, density, and cumulative distribution functions, Conditional Probability, Expected value,
Applications of the Univariate and Multivariate problems. Probabilistic inequalities, Random samples, sampling distributions of estimators and Maximum Likelihood.
Activities: Exercise based learning

UNIT II **14 Hours**
Statistical inference: Descriptive Statistics, Introduction to multivariate statistical models, Multivariate Regression, Multinominal regression and classification problems, principal components analysis. The problem of overfitting model assessment.
Introduction to Fuzzy Set Theory.
Activities: Analysis of live data from dataworld.org/Kaggle.com

UNIT III **16 Hours**
Graph Theory: Isomorphism, Planar graphs, graph colouring, Hamilton circuits and eulercycles.
Specialized techniques to solve combinatorial enumeration problems Graph Theory: Isomorphism, Planar graphs, graph colouring, Hamilton circuits and Euler cycles. Specialized techniques to solve combinatorial enumeration problems.
Activities: Simulation based learning from web resources

UNIT IV **14 Hours**
Computer science and engineering applications with any of following area: Data mining, Computer security, Software engineering, Computer architecture, Bioinformatics, Machine learning.

Recent Trends in various distribution functions in mathematical field of computer science for varying fields like, soft computing, and computer vision.
Activities: Problem solving and solution design of computer engineering problem.

Transactional Modes:

- Lecture
- Flipped Learning
- Collaborative Learning
- Peer Learning/Teaching
- Online Teaching Tools

Suggested Readings:

1. Vince, J. (2017). Foundation Mathematics for Computer Science. New York: Springer International Publishing.
2. Kishor, S. Trivedi. (2001). Probability and Statistics with Reliability, Queuing, and Computer Science Applications. United States: Wiley.
3. Mitzenmacher, M. and Upfal, E. (2005). Probability and Computing: Randomized Algorithms and Probabilistic Analysis. New Delhi: Cambridge University Press.
4. Tucker A. (2012). Applied Combinatorics. United States: Wiley.
5. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Code: CST.509
Course Title: Wireless Sensors Networks

Total Hours: 60

Course Objectives:
The Outcome of this course is to introduce students to the concepts of wireless sensor networks. That will help them to explain various MAC and routing protocols. The course will conclude with discussion on the security for possible attacks.

Course Outcomes
After completion of course, students would be able to:

- Describe and discuss various MAC and routing protocols.
- Employ and compare various MAC and routing protocols.
- Design wireless sensor networks in simulator.
- Evaluate the performance of various protocols using simulator.

UNIT I **16 Hours**
Introduction to Wireless Sensor Networks: Course Information, Introduction to Wireless Sensor Networks: Motivations, Applications, Performance metrics, History and Design factors.
Network Architecture: Traditional layered stack, Cross-layer designs, Sensor Network Architecture.

UNIT II **14 Hours**
Medium Access Control Protocol design: Fixed Access, Random Access, WSN protocols: synchronized, duty-cycled.
Introduction to Markov Chain: Discrete time Markov Chain definition, properties, classification and analysis.
MAC Protocol Analysis: Asynchronous duty-cycled. X-MAC Analysis (Markov Chain).

UNIT III **13 Hours**
Routing protocols for WSN: Resource-aware routing, Data-centric, Geographic Routing, Broadcast, Multicast
Opportunistic Routing Analysis: Analysis of opportunistic routing (Markov Chain).

UNIT IV **17 Hours**
Security: Possible attacks, countermeasures, SPINS, Static and dynamic key Distribution.
Introduction to Network Simulations: Introduction to Network Simulator, Description of the module and simulation example.
Advanced Topics: Recent development in WSN standards, software applications.

Transactional Modes:

- Lecture cum Demonstration
- Collaborative Learning
- E-tutorial
- Experimentation
- Online Teaching Tools

Suggested Readings:

1. Dargie, W., and Poellabauer, C. (2010). Fundamentals of Wireless Sensor Networks –Theory and Practice. United States: Wiley.
2. Sohraby, K., Minoli, D., and TaiebZnati. (2010). Wireless sensor networks – Technology: Protocols and Applications. United States: Wiley.
3. Hara, T., Vladimir, I.Z., and Buchmann, E., (2010). Wireless Sensor Network Technologies for the Information Explosion Era. New York: Springer.
4. Murthy, C.S. R. and Manoj B.S. (2004). Ad-hoc Wireless Networks Architectures and protocols. New Delhi: Pearson Education.
5. Obaidat M.S. and Misra, S. (2014). Principles of Wireless Sensor Networks. New Delhi: Cambridge University Press.
6. Misra, S., Woungang, I. and Misra S. C. (2009). Guide to Wireless Sensor Networks: Computer Communications and Networks Series. London: Springer.
7. He, J., Shouling, J., Pan, Y., and Yingshu, L. (2014). Wireless Adhoc and sensor networks. London: CRC press Taylor & Francis group.
8. Hu, F., Xiaojun, C. (2010). Wireless sensor networks. London: CRC press.
9. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.526
Course Title: Python Programing for Data Sciences

Total Hours: 60

Course Objectives:
The course introduce students to the Python programming language. It helps the students to handle object oriented problems with Python code and produce Python code to statistically analyse a dataset.

Course Outcomes
After completion of course, students would be able to:

- Define python environment and constructs of Python language.
- Apply Python language to construct scripts
- Analyse data with Python Libraries

UNIT I **15 Hours**

Python Introduction: Installing and setting Python environment in Windows and Linux, basics of Python interpreter, Execution of python program, Editor for Python code, syntax, variable, types. Flow control: if, if-else, for, while, range () function, continue, pass, break. Strings: Sequence operations, String Methods, Pattern Matching.
Activities: Implementation and solution of real time problem

UNIT II **15 Hours**

Lists: Basic Operations, Iteration, Indexing, Slicing and Matrixes; Dictionaries: Basic dictionary operations; Tuples: Basic Operations, Iteration, Indexing, Slicing; Functions: Definition, Call, Arguments, Scope rules and Name resolution;
Modules: Module Coding Basics, Importing Programs as Modules, Executing Modules as Scripts, Compiled Python files(.pyc), Standard Modules: OS and SYS, The dir() Function, Packages.
Activities: Assignment based Learning of real time problem

UNIT III **15 Hours**

Object Oriented Programming in Python: Classes, Objects, Inheritance, Operator Overloading, **File Handling:** Errors and Exceptions Handling (try and except) User-Defined Exception Objects.
Python Packages for Data Sciences: Mathematical and Statistical Analysis with NumPy, Manipulating and Visualisation of Data with SciPy
Activities: Analysis of real world data from Kaggle.com/dataworld.org website

UNIT IV **15 Hours**

Pandas: Shaping, merging, reshaping, slicing datasets and Data Structure, 2D Plot with matplotlib and seaborn.
Data Handling with Machine Learning: Use of Scikit packages for classification Problems, Linear regression, Logistic Regression, Decision Trees. Introduction of Tensor Flow for Natural and Deep Neural Network.
Activities: Statistical Modelling of various problems with machine learning

Transactional Modes:

- Lecture cum Demonstration
- Programme Learning
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Lutz, M., and Ascher, D. (2003). Learning Python. California: O'REILLY Media.
2. Berry, P. (2016). Head First Python, California: O'REILLY Media.
3. Jose, J., and Lal, P. S. (2016) Introduction to Computing & Problem Solving with Python. New Delhi: Khanna Books.
4. Lutz, Mark. (2012). Learning Python. New Delhi: Shroff publishers & distributors pvt. ltd.
5. Miller, Bradley N., Ranum, David L. (2014). Programming in context. Burlington: Jones & Bartlett learning.
6. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.510
Course Title: Compiler for HPC

Total Hours: 60

Course Objectives:

- To introduce the structure of compilers and high performance compiler design for students. Concepts of cache coherence and parallel loops in compilers are included.

Course Outcomes

After completion of course, students would be able to:

- Describe compiler structure.
- Discuss regarding parallel loops, data dependency and exception handling and debugging in compiler.
- Outline scalar, array region and concurrency analysis.
- Categorize and compare message passing machines

UNIT I **15 Hours**

High Performance Systems, Structure of a Compiler, Programming Language Features, Languages for High Performance.

Data Dependence: Data Dependence in Loops, Data Dependence in Conditionals, Data Dependence in Parallel Loops, Program Dependence Graph.

Scalar Analysis with Factored Use-Def Chains: Constructing Factored Use-Def Chains, FUD Chains for Arrays, Induction Variables Using FUD Chains, Constant Propagation with FUD Chains, Data Dependence for Scalars. Data Dependence Analysis for Arrays.

Activities: High performance Matrix Multiplication example , Discussion of various high performance computing systems

UNIT II **15 Hours**

Array Region Analysis, Pointer Analysis, I/O Dependence, Procedure Calls, Inter-procedural Analysis.

Loop Restructuring: Simple Transformations, Loop Fusion, Loop Fission, Loop Reversal, Loop Interchanging, Loop Skewing, Linear Loop. Transformations, Strip-Mining, Loop Tiling, Other Loop Transformations, and Inter-Procedural Transformations.

Optimizing for Locality: Single Reference to Each Array, Multiple References, General Tiling, Fission and Fusion for Locality.

Activities: Performing transformations such loop reversal , loop interchanging on Simple programs such as matrix multiplication

UNIT III **15 Hours**

Concurrency Analysis: Concurrency from Sequential Loops, Concurrency from Parallel Loops, Nested Loops, Round off Error, Exceptions and Debuggers.

Vector Analysis: Vector Code, Vector Code from Sequential Loops, Vector Code from for all Loops, Nested Loops, Round off Error, Exceptions, and Debuggers, Multi-vector Computers.

Activities: Group discussion and presentation by the students

UNIT IV **15 Hours**

Message-Passing Machines: SIMD Machines, MIMD Machines, Data Layout, Parallel Code for Array Assignment, Remote Data Access, Automatic Data Layout, Multiple Array Assignments, Other Topics.

Scalable Shared-Memory Machines: Global Cache Coherence, Local Cache Coherence, Latency Tolerant Machines.

Recent trends in compiler design for high performance computing and message passing machines and scalable shared memory machine.

Activities: Group Discussion of Survey papers on SIMD, MIMD architectures

Transactional Modes:

- Lecture cum Demonstration
- Peer Learning/Teaching
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Wolfe, M. (1995). High-Performance Compilers for Parallel Computing. New Delhi: Pearson Education.
2. Muchnick, S. (1997). Advanced Compiler Design and Implementation. Elsevier.
3. Allen. (2001). Optimizing Compilers for Modern Architectures. California: Morgan Kaufmann.
4. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Code: CST. 511
Course Title: Distributed Database System

Total Hours: 60

Course Objectives:
To introduce the fundamental concepts and issues of managing large volume of shared data in a parallel and distributed environment. Provide insight into related research problems.

Course Outcomes
After completion of course, students would be able to:
- Explain trends in distributed systems.
- Demonstrate distributed query optimization.
- Examine distributed system design and query processing issues.
- Categorize and assess reliability issues in distributed systems.

UNIT I **15 Hours**
Introduction: Distributed data processing; what is a DDBS; Advantages and disadvantages of DDBS; Problem areas; Overview of database and computer network concepts.
Distributed Database Management System Architecture: Transparencies in a distributed DBMS; Distributed DBMS architecture; Global directory issues.
Activities: Example based study of Distributed Database Systems

UNIT II **15 Hours**
Distributed Database Design: Alternative design strategies; Distributed design issues; Fragmentation; Data allocation.
Semantics Data Control: View management; Data security; Semantic Integrity Control.
Query Processing Issues: Objectives of query processing; Characterization of query processors; Layers of query processing; Query decomposition; Localization of distributed data.
Activities: Discussion of Distributed Database Design using Case Studies from good journal papers.

UNIT III **15 Hours**
Distributed Query Optimization: Factors governing query optimization; Centralized query optimization; ordering of fragment queries; Distributed query optimization algorithms.
Transaction Management: The transaction concept; Goals of transaction management; Characteristics of transactions; Taxonomy of transaction models.
Concurrency Control: Concurrency control in centralized database systems; Concurrency control in DDBSs; Distributed concurrency control algorithms; Deadlock management.
Activities: Discussion of Survey papers on Concurrency control techniques (optimistic vs pessimistic) in centralized database systems and Distributed systems, Example based study of query optimisation techniques

UNIT IV **15 Hours**

Reliability: Reliability issues in DDBSs; Types of failures; Reliability techniques; Commit Protocols; Recovery protocols.

Parallel Database Systems: Parallel architectures; parallel query processing and optimization; load balancing.

Introduction to cloud computing, Advanced Topics: Mobile Databases, Distributed Object Management, Multi-databases.

Activities: PowerPoint presentations by students

Transactional Modes:

- Lecture cum Demonstration
- Case study
- E-tutorial
- Collaborative Learning
- Online Teaching Tools

Suggested Readings:

1. Ozsu, M.T., and Valduriez, P. (2011). Principles of Distributed Database Systems, United States: Prentice-Hall.
2. Bell D., and Grimson, J., (1992). Distributed Database Systems. United States: Addison-Wesley.
3. Deshpande, S., (2014). Distributed Databases. New Delhi: Dreamtech Press.
4. Saeed, K. R., Frank, S. H. (2010). Distributed Database Management Systems: A Practical Approach. New Delhi: Wiley.
5. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.512
Course Title: Information Security

Total Hours: 60

Course Objectives:
To introduce students to the concept of security, and types of attack. Help students to understand Symmetric & Asymmetric Key Cryptography. The course will also give exposure on Internet Security Protocol.

Course Outcomes
After completion of course, students would be able to:
- Identify the domain specific security issues.
- Apply Symmetric & Asymmetric Key Cryptography in various applications.
- Analyze Internet Security Protocols.
- Compare and contrast various internet security protocols

UNIT I **12 Hours**
History of Information Systems: Importance of Information Systems, Basics of Information Systems, Changing Nature of Information Systems, Global Information Systems.
Introduction to Security: Need for security, Security Trends, Security Attacks, Security Services, and Security Mechanisms. Security techniques.
Activities: Group Discussion**, Case** study based learning of different information system and cybercrimes.

UNIT II **16 Hours**
Mathematics of Cryptography- Prime and Composite Numbers, Greatest Common Divisor, Euclidean algorithm, Modulo arithmetic, Fermat's little theorem, Multiplicative Inverse, Euler's theorem and Totient function, Discrete logarithm.
Classical Cryptographic Algorithms: Substitutions techniques- Monoalphabetic ciphers, Polyalphabetic Ciphers, Transposition Techniques, Rotor Machines, Cryptanalysis of classical cryptographic algorithms
Activities: Assignment based and numerical exercise based learning, Implementation of various cryptographic algorithms using computer programming.

UNIT III **16 Hours**
Morden symmetric key Cryptographic Algorithms: Data Encryption Standard (DES), Triple DES, IDEA, Advance Encryption Algorithm (AES), Differential & Linear Cryptanalysis.
Asymmetric key Cryptographic Algorithms:- Public-Key Cryptography Principles, Diffie–Hellman key exchange algorithm, Knapsack algorithm, RSA.
Message Authentication: Approaches to Message Authentication, MD5, SHA-512, Digital Signature Standard (DSS).
Activities: Implementation and web based simulation of various cryptographic algorithms.

UNIT IV **16 Hours**

Network Security Protocol: Introduction, Security at the Application Layer: PGP and S/MIME, Secure Electronic Transaction, 3-D Secure Protocol, Security at the Transport Layer: Secure Socket Layer (SSL), Transport Layer Security (TLS), Security at the Network Layer: IPSec, Wireless Application Protocol (WAP) Security. Wired Equivalent Privacy (WEP).
Activities: Case study of various network security protocols, Brainstorming, Implementation and solution of real time cryptographic problems.

Transactional Modes:

- Lecture cum Demonstration
- Peer Learning/Teaching
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Forouzan, B. A., & Mukhopadhyay, D. (2015). Cryptography & Network Security. New Delhi: Tata McGraw-Hill Education.
2. Kahate, A. (2017). Cryptography and Network Security. New Delhi: Tata McGraw-Hill Education.
3. Godbole, N. (2017). Information Systems Security: Security Management, Metrics, frameworks and Best Practices. New Delhi: John Wiley & Sons India.
4. Stallings, W. (2011). Network Security Essentials: Applications and standards. New Delhi: Pearson Education India.
5. Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. New Delhi: Pearson Education India.
6. Kim, D., & Solomon, M. G. (2016). Fundamentals of Information Systems Security. Massachusetts: Jones & Bartlett Publishers.
7. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.513
Course Title: Software Testing & Maintenance

Total Hours: 60

Course Objectives:
To enable a clear understanding and knowledge of the foundations, techniques, and tools in the area of software testing and its practice in the industry. To identify the software testing process for software quality checking. The help students design metrics models for predicting software testing and maintenance requirements.

Course Outcomes
After completion of course, students would be able to:

- Apply software testing knowledge, verification & validation and engineering methods.
- Design and conduct a software test process for a quality software test.
- Identify various software testing problems, and solve these problems by designing and selecting software metrics models, testing criteria, strategies, and methods.

UNIT I **14 Hours**

Overview of Software Engineering: Phases in development of Software, Software Engineering Ethics, Life cycle Revisited (Incremental Development, Agile Methods, RAD), Model-Driven Architecture, Software Product Line, Process Modelling**.**
Project Management: Project Planning, Project Control (Work Break Down Structure, GANTT Charts, PERT Charts) Project Team Organization, Risk Management, CMM.
Activities: Assignment base learning and Group Discussion

UNIT II **15 Hours**

Testing of OO systems: Objects and Classes, OO Testing, Class Testing, Regression Testing, Non-Functional Testing, Acceptance Testing, Mutation Testing.
Software Testing: Levels of testing, Module, Integration, System, Regression, Testing techniques and their Applicability, Functional testing and Analysis Structural testing and Analysis, Error Oriented testing and Analysis, Hybrid Approaches, Integration Strategies, Transaction Flow Analysis, Stress Analysis, Failure Analysis, Concurrency Analysis.
Activities: Case study of various organisation projects

UNIT III **15 Hours**

Overview of Software Metrics: Measurement in Software Engineering, Scope of Software Metrics, Measurement and Models Meaningfulness in measurement, Measurement quality, Measurement process, Scale, Measurement validation, Object-oriented measurements.
Measuring Internal External Product Attributes: Measuring size, aspects of software size, length, functionality and complexity, measuring structure, types of

structural measures, Modeling software quality, measuring aspects of software quality, software reliability, basics of software reliability.
Activities: Case study of various organisation projects

UNIT IV **16 Hours**

Software Maintenance: Maintenance Categories, Major causes of Maintenance Problems, Reverse Engineering, Software Evolutions, Organizational and Managerial Issues of Maintenance activities, Maintenance Measurements Software Refactoring: Principles of Refactoring, Bad Smells in code, Composing Methods of Refactoring, Moving features between objects.
Activities: Statistical Modelling of maintenance and Refactoring

Transactional Modes:

- Lecture
- Peer Learning/Teaching
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Pressman, R. S. (2017). Software Engineering a Practitioners Approach. New Delhi: McGraw Hill Education India Private Limited.
2. Peters, J. S., and Pedrycz, W. (2007). Software engineering an engineering approach. New Delhi: Wiley India.
3. Basu A. (2015). Software Quality Assurance, Testing and Metrics. New Delhi: PHI India.
4. Vliet, H.V. (2008). Software Engineering Principles and Practice. United States: John Wiley & Sons.
5. Ghezzi, C., Jazayeri, M., and Mandriolo, D. (2012). Fundamental of Software Engineering, New Delhi: PHI Private limited.
6. Mall, R. (2011). Fundamentals of Software Engineering. New Delhi: PHI learning.
7. Singh, Y., Aggarwal, K.K. (2014). Software engineering, New Delhi: New age international publishers.
8. Sommerville, I. (2014). Software engineering. New Delhi: Pearson education.
9. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.606
Course Title: Research Methodology and IPR

Total Hours: 60

Course Objectives:
To develop a research orientation among the students and help them understand fundamentals of research methods. The course will help the students to identify various sources of information for literature review, data collection and effective paper/ dissertation writing. Familiarize students with the concept of patents and copyright

Course Outcomes
After completion of course, students would be able to:

- Explain effective methods to formulate a research problem.
- Analyze research related information and follow research ethics.
- Apply intellectual property law principles (including copyright, patents, designs and trademarks) to practical problems and be able to analyse the social impact of IPR.

UNIT I **14 Hours**

Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations.
Activities: Assignment based learning

UNIT II **15 Hours**

Effective literature studies approaches, analysis Plagiarism, Research ethics,
Effective technical writing, how to write report, Paper
Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee.
Activities: Analysis of various tools and Case Studies

UNIT III **14 Hours**

Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development.
International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.
Activities: Case Studies

UNIT IV **16 Hours**

Patent Rights: Scope of Patent Rights. Licensing and transfer of technology. Patent information and databases. Geographical Indications.
New Developments in IPR: Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software, Integrated Circuits, etc.

Activities: Group discussion

Transactional Modes:

- Lecture
- Case Studies
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Melville, S., and Goddard, W. (1996). Research methodology: An introduction for science & engineering students. South Africa: Juta Academic.
2. Goddard, W., and Melville, S. (2001). Research Methodology: An Introduction. South Africa: Juta Academic.
3. Kumar, R. (2019). Research Methodology: A Step by Step Guide for beginners. New Delhi: SAGE Publications Ltd.
4. Halbert, (2006). Resisting Intellectual Property. New Delhi: Taylor & Francis Ltd.
5. Mayall, (2011). Industrial Design. New Delhi: McGraw Hill.
6. Niebel, (1974). Product Design. New Delhi: McGraw Hill.
7. Asimov, M. (1976). Introduction to Design. United States: Prentice Hall.
8. Merges, R. P., Menell, P. S., & Lemley, M. A. (2003). Intellectual Property in New Technological Age. United States: Aspen Law & Business.
9. Flick, U. (2011). Introducing research methodology: A beginner's guide to doing a research project. New Delhi: Sage Publications India.
10. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
0	0	2	1

Course Code: CST.515
Course Title: Advanced Data Structures – Lab

Course Objectives:
The lab is designed to help students develop skills to design and analyse advance data structures. To help students identify and apply the suitable data structure for a given problem.

Course Outcomes
After completion of course, students would be able to:
- Design and analyse different data structures.
- Choose the appropriate data structure for a given problem.

Lab Assignments

1. Write a program to find Factorial using Recursion & Iteration
2. Write a program to implement hashing with chaining
3. Write a program to implement 2 dimensional Array
4. Write a program to create Single Linked List of integers.
5. Write a program to implement Bubble Sort
6. Write a program to implement Quicksort
7. Write a program to implement Merge Sort
8. Write a program to create a Binary Search Tree
9. Write a program to implement Heap Sort
10. Write a program to implement Skip List
11. Write a program to perform insertion, deletion and traversal In B Tree

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

Suggested Readings:

1. Lab Manual
2. Allen Weiss M. (2014). Data Structures and Algorithm Analysis in C++. New Delhi: Pearson Education.

L	T	P	Cr
0	0	2	1

Course Code: CST.516
Course Title: Wireless Sensors Networks Lab

Course Objectives:
The objective of this course is to introduce students to the difference between wired and wireless networks. Help them to differentiate between various protocols. Describe the various security loopholes and their countermeasures in wireless sensor networks.

Course Outcomes
After completion of course, students would be able to:

- Design the Wired and Wireless networks using suitable tools.
- Analyze the wireless sensor networks using various protocols.
- Evaluate the performance of sensor networks.

Suggested Readings:

1. Lab Manual
2. Sohraby, K., Minoli, D., and TaiebZnati. (2010). Wireless sensor networks – Technology: Protocols and Applications. United States: Wiley.

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

L	T	P	Cr
0	0	4	2

Course Code: CST.528
Course Title: Python Programming for Data Sciences Lab

Course Objectives:
To help students understand the basic constructs of Python Interpreter. To demonstrate the working of Python functions and modules w.r.t definition call and scope. To make students acquainted with OOPS and File handling concept in Python and to understand and apply various Python packages for Data handling.

Course Outcomes:
After Completion of the lab course the students will be able:

- To create and demonstrate script in Python by using basic constructs and control statements of Python.
- To illustrate the use of OOPS and file handling concept for data handling and visualisation.
- To synthesize the code in Python by using various Data Handling libraries.

Students will implement the following lab practical's

List of Practicals:

1. Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5,between 2000 and 3200 (both included).The numbers obtained should be printed in a comma-separated sequence on a single line.
2. Write a program which can compute the factorial of a given numbers.The results should be printed in a comma-separated sequence on a single line.Suppose the following input is supplied to the program: 8 .Then, the output should be:40320
3. With a given integral number n, write a program to generate a dictionary that contains (i, i*i) such that is an integral number between 1 and n (both included). and then the program should print the dictionary.

   Suppose the following input is supplied to the program:
   8 Then, the output should be:
   {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}

4. Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number.

   Suppose the following input is supplied to the program: 34,67,55,33,12,98

Then, the output should be:

['34', '67', '55', '33', '12', '98']

('34', '67', '55', '33', '12', '98')

5. Define a class which has at least two methods: getString: to get a string from console input printString: to print the string in upper case.
   Also please include simple test function to test the class methods.

6. Write a program that calculates and prints the value according to the given formula:

   Q = Square root of [(2 * C * D)/H]

   Following are the fixed values of C and H: C is 50. H is 30. D is the variable whose values should be input to your program in a comma-separated sequence.

   Example:

   Let us assume the following comma separated input sequence is given to the program: 100,150,180

   The output of the program should be: 18,22,24

7. Write a program which takes 2 digits, X,Y as input and generates a 2- dimensional array. The element value in the i-th row and j-th column of the array should be i*j. Note: i=0,1.., X-1; j=0,1,¡•Y-1.

   Example: Suppose the following inputs are given to the program: 3,5 Then, the output of the program should be:
   [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]

8. Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically.

   Suppose the following input is supplied to the program: without, hello, bag, world

   Then, the output should be: bag, hello, without, world

9. Write a program that accepts sequence of lines as input and prints the lines after making all characters in the sentence capitalized. Suppose the following input is supplied to the program:

   - Hello world

   - Practice makes
   perfect Then, the output
   should be:

- HELLO WORLD
- PRACTICE MAKES PERFECT

10. Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically. Suppose the following input is supplied to the program:

    - hello world and practice makes perfect and hello world again Then, the output should be:
    - again and hello makes perfect practice world

11. Write a program which accepts a sequence of comma separated 4 digit binary numbers as its input and then check whether they are divisible by 5 or not. The numbers that are divisible by 5 are to be printed in a comma separated sequence.

    Example:

    0100,0011,1010,1001

    Then output should be:1010

    Notes: Assume the data is input by console.

12. Write a program that computes the value of a+aa+aaa+aaaa with a given digit as the value of a.

    Suppose the following input is supplied to the program: 9
    Then, the output should be:
    11106

13. Write a program, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even number.

    The numbers obtained should be printed in a comma-separated sequence on a single line.

14. Write a program that accepts a sentence and calculate the number of letters and digits.

    Suppose the following input is supplied to the program: hello world! 123
    Then, the output should be:

    LETTERS 10. DIGITS 3

15. Write a program that accepts a sentence and calculate the number of upper case letters and lower case letters.

    Suppose the following input is supplied to the program: Hello world!
    Then, the output should be:

    UPPER CASE 1

    LOWER CASE 9

16. Use a list comprehension to square each odd number in a list. The list is input by a sequence of comma-separated numbers.

    Suppose the following input is supplied to the program: 1,2,3,4,5,6,7,8,9
    Then, the output should be:

    1,3,5,7,9

17. Write a program that computes the net amount of a bank account based a transaction log from console input. The transaction log format is shown as following:

    D 100

    W 200

    D means deposit while W means withdrawal.

    Suppose the following input is supplied to the program: D 300

D 300

W 200

D 100

Then, the output should be:

500

18. A website requires the users to input username and password to register. Write a program to check the validity of password input by users.

    Following are the criteria for checking the password:

    1. At least 1 letter between [a-z]

    2. At least 1 number between [0-9]

    1. At least 1 letter between [A-Z]

3. At least 1 character from [$#@]

4. Minimum length of transaction password: 6

5. Maximum length of transaction password: 12

Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma.

Example

If the following passwords are given as input to the program:
ABd1234@1,a F1#,2w3E*,2We3345
Then, the output of the program should be:

ABd1234@1

19. You are required to write a program to sort the (name, age, height) tuples by ascending order where name is string, age and height are numbers. The tuples are input by console. The sort criteria is:

1: Sort based on name;

2: Then sort based on
age; 3: Then sort by
score.

The priority is that name > age > score.

If the following tuples are given as input to the program:
Tom,19,80
John,20,9
0
Jony,17,9
1
Jony,17,9
3
Json,21,8
5
Then, the output of the program should be:

[('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json', '21', '85'), ('Tom', '19', '80')]

20. Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n.

    Hints: Consider use yield

Lab Evaluation:

The evaluation of lab criteria will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

Suggested Readings:

Lab Manual

L	T	P	Cr
0	0	2	1

Course Code: CST.518
Course Title: Complier for HPC Lab

Course Objectives:
The course is designed to help students apply the Concepts like instruction level, data level and thread level parallelism. The students will be able to design, implement and analyse the parallel programs on shared memory and distributed memory systems.

Course Objectives:
After the completion of the course the students will be able to

- Identify some common machine independent optimizations.
- Apply Compiler techniques and tools for exploiting instructions, data and thread level parallelism.
- Evaluate memory locality optimizations.

Suggested Readings:

1. Lab Manual
2. Wolfe, M. (1995). High-Performance Compilers for Parallel Computing. New Delhi: Pearson Education.

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	30
End Term (Implementation and Viva-Voce)	20
Total	**50**

L	T	P	Cr
0	0	2	1

Course Code: CST.514
Course Title: Distributed Database System Lab

Course Objectives:
The objective of this course is to:

- To introduce the basic concepts and implementation methods of Distributed Database systems.
- To uncover trending research issues in Distributed Database systems.
- To develop various applications related to Distributed Database systems.
- To put theory to practice by building and furnishing a distributed database query engine, subject to remote Web service calls.

Course Outcomes
After completion of course, students would be able to:

- Develop practical skills in the use of approaches for Distributed Database systems.
- Select and apply the appropriate approach for a particular case.
- Apply learned skills for solving practical database related tasks.
- Produce the transaction management and query processing techniques in Distributed Database systems.

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

Suggested Readings:

Lab Manual

L	T	P	Cr
0	0	2	1

Course Code: CST.519
Course Title: Information Security Lab

Course Objectives:
To introduce students to the concept of security, and types of attack. Help students to understand Symmetric & Asymmetric Key Cryptography. The course will also give exposure on Internet Security Protocol.

Learning Outcomes

- Identify the domain specific security issues.
- Implement Symmetric & Asymmetric Key Cryptography in various applications.
- Analyze Internet Security Protocols.

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

Suggested Readings:

1. Lab Manual
2. Forouzan, B. A., & Mukhopadhyay, D. (2015). Cryptography & Network Security. New Delhi: Tata McGraw-Hill Education.
3. Kahate, A. (2017). Cryptography and Network Security. New Delhi: Tata McGraw-Hill Education.

L	T	P	Cr
0	0	2	1

Course Code: CST.520
Course Title: Software Testing & Maintenance Lab

Course Objectives:
To learn and apply the tools in the area of software testing and its practice in the industry. To apply the software testing process for software quality checking and assurance. To design metrics models for predicting software testing and maintenance requirements.

Course Outcomes
After completion of course, students would be able to:

- Apply software testing techniques for verification & validation of software.
- Design and conduct a software test process for a quality checking and assurance.
- Identify software metrics models, testing criteria, strategies, and methods.

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

Suggested Readings:
Lab Manual
.

Interdisciplinary Course (IDC) Semester-I

L	T	P	Cr
2	0	0	2

Course Code: CBS.518
Course Title: IT Fundamentals

Total Hours: 30

Course Outcomes

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

- Describe different hardware and software components of computer.
- Use word processing, presentation and spreadsheet software.
- Illustrate the concept of networking and internet.

UNIT I **8 Hours**

Fundamentals of Computers: Parts of computers, Hardware, BIOS, Operating systems, Binary system, Logic gates and Boolean Algebra. Introduction to computer network and World Wide Web, Storage space, CPU and Memory.
Activities: Numerical Based exercises for conversion of Binary to octal , hexadecimal and decimal number system , Identification of various ports by the students on such as Audio ports, USB ports, HDMI Port, Ethernet port

UNIT II **7 Hours**

MS-Word: Introduction to Word Processing, Creating and Saving Documents, Text Formatting, Tables, Document Review Option, Mail Merge, Inserting Table of Contents, Reference Management.
Activities: Error free typing exercises, Insertion of in text citations and insertion of Bibliography at the end of the document, Insertion of Tables and figures and cross referencing them from the text.

UNIT III **8 Hours**

Applications Software: Introduction to MS Paint, Notepad, Spreadsheet applications, Presentation applications, Internet browsers and Image processing applications.
Activities: Creation of a powerpoint presentation by students with various animation and and transition effects, Creation of an excel workbook by the students and application of basic mathematical functions (such as sum, average, Count, Mean, Median , Mode) on the data

UNIT IV **7 Hours**

World Wide Web: Origin and concepts, Latency and bandwidth, searching the internet, Advanced web-search using Boolean logic, Networking fundamentals.
Activities: searching for some relevant articles using keyword combinations on various electronic databases using advanced search options by students

Transactional Modes:

- Lecture
- Blended Learning

- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Gookin, D. (2007). MS Word for Dummies. United States: Wiley.
2. Harvey, G. (2007). MS Excel for Dummies. United States: Wiley
3. Sinha, P.K. (2004). Computer Fundamentals. New Delhi: BPB Publications.
4. Bott, E. (2009). Windows 7 Inside Out. United States: Microsoft Press.
5. Goel, A., Ray, S. K. (2012). Computers: Basics and Applications. New Delhi: Pearson Education India.
6. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
2	0	0	2

Course Code: CBS.519
Course Title: Programming in C

Total Hours: 30

Course Outcomes
At the end of this course, students will be able to:
- Describe the concept and need of programming.
- Explain syntax and use of different functions available in C.
- Demonstrate programming in C.

UNIT I **8 Hours**
Introduction to Programming Language: Types of Programming Language, Structured Programming, Algorithms and Flowcharts, Programming Language.
Introduction to C: History, Character Set, Structure of a C Program – constants, variables and Keywords, data types, expression statements, compound statements.
Activities: Program Fragments based exercises to find out output of various program fragments using the studied concepts

UNIT II **8 Hours**
C Operators: Arithmetic, Unary, Relational and Logical, Assignment, Conditional Operator, Increment, decrement Operator, Using library function in math.
Data Input Output: Single character input, getchar, getch, getc, single character output putchar, putc, Formatted I/O.
Activities: Program Fragments based exercises

UNIT III **7 Hours**
C Constructs: If statement, while statement, do….while statement, for statement, switch statement, nested control statement, break, continue, goto statement.
C Functions: Functions, Definiton and scope, Assessing and Prototyping, Types of functions, Passing arguments to functions.
Activities: Program fragments based exercises, Creating User defined function to perform simple activities and using them in C program

UNIT IV **7 Hours**
Arrays and Strings: Single dimensional array, Multi-dimensional array, Initializing array using static declaration, character array and strings, String Handling functions.
Activities: Program fragment based exercises, Pseudocode to implement single and multi dimensional arrays concept for practical programs

Transactional Modes:
- Lecture
- Blended Learning
- E-tutorial
- Self-Learning

- Online Teaching Tools

Suggested Readings:

1. Rajaraman, V. (2008). Computer Basics and C Programming PHI Learning.
2. Brown, T. D. (1987) C for Basic Programmers. United States: Silicon Press.
3. Kanetkar, Y. P. (2010). Let Us C. New Delhi: BPB Publications.
4. Balagurusamy. (2008). Programming in ANSI C. New Delhi: Tata Mcgraw-Hill.
5. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
2	0	0	2

Course Code: CST.530
Course Title: Introduction to Digital Logic

Total Hours: 30

Course Outcomes
At the end of this course, students will be able to:
- Describe the digital signal along with the operations applicable on them.
- Discuss different number systems and conversion between them along with memory devices used to store such data.
- Apply the Boolean laws in different situation.

UNIT I **8 Hours**
Introduction: Digital Signals, basic digital circuits: AND operation, OR operation and NOT operation.
Number Systems: Introduction, Binary number system, Octal number system, Hexadecimal Number system, Conversion of one number system to other, Gray code.
Activities: Web based Simulation learning

UNIT II **7 Hours**
Logic Gates and Boolean Algebra: Boolean Laws, Boolean expression and functions, Logic Gates.
Activities: Web based Simulation learning

UNIT III **8 Hours**
Combinational Circuit Design: Karnaugh Map representation of logic functions, SOP, POS, Simplification of logic functions using K-Map.
Activities: Exercise based learning

UNIT IV **7 Hours**
Flip-Flops: 1-bit memory cell, S-R Flip Flop, J-K Flip Flop, D- Flip Flop, T- Flip Flop.
Activities: Web based simmulation

Transactional Modes:
- Lecture
- Blended Learning
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Mano, M. and Charles, K. (2007). Logic and Computer Design Fundamentals. New Delhi: Pearson Education.
2. Jain, R.P. (2006). Modern Digital Electronics. New Delhi: Tata McGraw Hill.
3. Kharate, G.K. (2010). Digital Electronics. United States: Oxford Higher Education.
4. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
2	0	0	2

Course Code: CST.531
Course Title: Multimedia and Its Applications

Total Hours: 30

Course Outcomes
At the end of this course, students will be able to:
- Identify and analyze different types of multimedia along with their representation.
- Differentiate between formats of all types of multimedia.
- Plan where we can use these multimedia.

UNIT I **8 Hours**
Introductory Concepts: Multimedia-Definitions, Basic properties and medium types. Multimedia applications, Uses of Multimedia.
Sound/ Audio: Basic Sound Concepts, Music. **Speech:** Generation, Analysis and Transmission.
Activities: Group Discussion

UNIT II **7 Hours**
Images and Graphics: Basic concepts: Image representation, image format, Graphics Format, Computer Image Processing.
Video and Animation: Basic Concepts: Video Signal Representation, Computer Video Format. Television: Conventional Systems, Enhanced Definition Systems, High-Definition Systems.
Activities: Web based learning

UNIT III **7 Hours**
Data Compression: Storage space, coding requirements, JPEG, MPEG.
Miscellaneous: Optical Storage Media, Mutlimedia Operating Systems, Multimedia Communication Systems.
Activities: Simulation based Learning

UNIT IV **8 Hours**
Documents and Hypertext: Document Architecture, Manipulation of Multimedia Data, Hypertext, Hypermedia and Multimedia and example.
Multimedia Applications: Media Preparation, composition, Integration, communication, Consumption, and Entertainment.
Activities: Group Discussion

Transactional Modes:
- Lecture
- Blended Learning
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Steinmetz, R. (2009). Multimedia: Computing Communications & Applications. New Delhi: Pearson Education India.
2. Vaughan, T. (2008). Multimedia: making it work. New Delhi: Tata McGraw-Hill Education.
3. Rao, K.R., Bojkovic, Z. S. and Milovanovic, D. A. (2002). Multimedia Communication Systems: Techniques, Standards, and Networks. United States: Prentice Hall.
4. Andleigh, P.K. (2007). Multimedia Systems Design. United States: Prentice Hall
5. Rimmer, S. (2007). Advanced Multimedia Programming. New Delhi: Windcrest/McGraw-Hill.
6. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
2	0	0	2

Course Code: CBS.532
Course Title: Introduction to MATLAB

Total Hours: 30

Course Outcomes
At the end of this course, students will be able to:
- Describe the basic syntax of MATLAB along with various functions available in it.
- Analyze all the functions in graphical manner.
- Design a GUI interface for any software.

UNIT I **8 Hours**
Introduction: MatLab, MatLab Syntax and interactive computations.
Live Demonstration of MATLAB command prompt
Activities: Assignment based learning

UNIT II **7 Hours**
Programming: in Matlab using procedures and functions: Arguments and return values, M-files, Formatted console input-output, String handling.
Live Demonstration of MATLAB M-files
Activities: Assignment based learning

UNIT III **8 Hours**
Control Statements: Conditional statements: If, Else, Elseif. Repetition statements: While, For.
Manipulating Text: Writing to a text file, Reading from a text.
Activities: Creation of text files and assignment based learning

UNIT IV **7 Hours**
Graph Plots: Basic plotting, Built in functions
GUI Interface: Attaching buttons to actions, Getting Input, Setting Output Using the toolboxes.
Activities: Creation of GUI relevant to the departments.

Transactional Modes:
- Lecture
- Blended Learning
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Attaway. (2012). Matlab: A Practical Introduction to Programming and Problem Solving. Elsevier
2. Pratap, R. (2010). Getting Started with MATLAB: A Quick Introduction for Scientists and Engineers. New Delhi: Oxford.
3. Research Articles from SCI & Scopus indexed Journals.

SEMESTER -II

L	T	P	Cr
4	0	0	4

Course Code: CST.521
Course Title: Advance Algorithms

Total Hours: 60

Course Objectives:
To familiarize students with basic paradigms and data structures used to solve advanced algorithmic problems. To introduce the students to recent developments in the area of algorithmic design.

Course Outcomes
After completion of course, students would be able to:

- Analyze the complexity/performance of different algorithms.
- Identify the appropriate data structure for solving a particular set of problems.
- Categorize the different problems in various classes according to their complexity.

UNIT I **16 Hours**

Sorting: Review of various sorting algorithms, topological sorting
Graph: Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkasra's), depth-first search and computation of strongly connected components,
Emphasis on correctness proof of the algorithm and time/space analysis,
Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set. Application to MST.
Activities: Assignment based learning, visual modelling and web based animation of different algorithms (VisuAlgo Project).

UNIT II **14 Hours**

Strassen's algorithm and introduction to divide and conquer paradigm, inverse of a triangular matrix, relation between the time complexities of basic matrix operations.
Floyd-Warshall algorithm and introduction to dynamic programming paradigm. More examples of dynamic programming.
Activities: Brainstorming, assignment based learning and implementation of different algorithms.

UNIT III **14 Hours**

Linear Programming: Geometry of the feasibility region and Simplex algorithm,
Decision Problems: P, NP, NP Complete, NP-Hard,
NP Hard with Examples, Proof of NP-hardness and NP-completeness.
Activities: Problem solving and solution design, Exercise based learning, open book assignment.

UNIT IV **16 Hours**

One or more of the following topics based on time and interest Approximation algorithms, Randomized Algorithms, Interior Point Method,
Recent Trends in problem solving paradigms using recent searching and sorting techniques by applying recently proposed data structures.
Activities: Student presentation, Case study of different NP class problems and solution design using Randomised algorithms and Approximation Algorithms.

Transactional Modes:

- Lecture
- Blended Learning
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Cormen, T. H., Leiserson, C. E., and Rivest, P. L. (2010). Introduction to Algorithms. Cambridge: MIT Press.
2. Aho, A. V., Hopcroft, J. E., and Ullman, J. D. (2002). The Design and Analysis of Computer Algorithms. New Delhi: Pearson Education India.
3. Kleinberg, J., and Tardos. E. (2005). Algorithm Design. New Delhi: Pearson Education India.
4. Hromkovic, J. (2015). Design and Analysis of Randomized Algorithms: Introduction to Design Paradigms. New York: Springer.
5. Baase, S., Gelder V., and Allen. (2009) Computer algorithms: introduction to design & analysis. New Delhi: Pearson Education.
6. Benoit, Anne, Robert, Yves, Vivien, and Frederic. (2014). A guide to algorithm design: Paradigms, methods and complexity analysis, London: CRC Press Taylor & Francis group.
7. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.522
Course Title: Soft Computing

Total Hours: 60

Course Objectives:
To introduce the students to soft computing concepts and techniques and foster their abilities in designing appropriate technique for a given scenario. To give students knowledge with hands-on experience of non-traditional technologies and fundamentals of artificial neural networks, fuzzy sets, fuzzy logic, genetic algorithms.

Course Outcomes
After completion of course, students would be able to:

- Identify and describe soft computing techniques and their roles in building intelligent machines.
- Apply fuzzy logic and reasoning to handle uncertainty and solve various engineering problems.
- Apply genetic algorithms to optimization problems.
- Evaluate and compare solution using various soft computing approaches for a given problem.

UNIT I **Hours: 14**

Introduction to Soft Computing: Evolution of Computing: Soft Computing Constituents, From Conventional, Major areas of Soft Computing, applications of Soft Computing.

Activity: Brainstorming, assignment based learning
Neural Networks: Introduction, Brief history, Neural Networks Characteristics, architecture, and properties.
Neural Network Learning Algorithm Machine Learning Using Neural Networks.
Activities: Exercise based learning and practical hands on training

UNIT II **Hours: 16**

Fuzzy Logic: Fuzzy Sets, Membership Functions, Operations on Fuzzy Sets, Fuzzy Relations.
Fuzzy Rules and Fuzzy Reasoning, Fuzzy Inference Systems, Fuzzy Expert Systems, Fuzzy Decision Making, Fuzzy Models.
Activities: Assignment and Exercise based learning

UNIT III **Hours: 14**

Genetic Algorithms: Introduction to Genetic Algorithms (GA), Applications of GA in Machine Learning: Machine Learning Approach to Knowledge Acquisition. Introduction to other optimization techniques.
Activities: Exercise and case study based learning

UNIT IV **Hours: 16**

Swarm intelligence: Overview, mechanism, technologies like particle swarm optimization, ant colony optimization, cuckoo search.

Introduction to hybrid systems: Neuro Fuzzy, Neuro Genetics and Fuzzy Genetic system.
Recent trends in soft computing techniques.
Activities: Student presentation and group discussion

Transactional Modes:

- Lecture cum Demonstration
- Peer Learning/Teaching
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Jang, J. R. S., Sun, C. T., and Mizutani E. (1997). Neuro - Fuzzy and Soft Computing, New Delhi: Prentice-Hall of India, Pearson.
2. Klir, G. J., and Yuan, B. (2015). Fuzzy Sets and Fuzzy Logic - Theory and Applications. New Delhi: Pearson Education India.
3. Ross, J. T. (2011). Fuzzy Logic with Engineering Applications. United States: John Wiley & Sons.
4. Rajasekaran, S., and Vijayalakshmi Pai, G.A. (2013). Neural Networks, Fuzzy Logic and Genetic Algorithms. United States: Prentice Hall India Learning.
5. Priddy, K. L., and Keller, E. P. (2005). Artificial Neural Networks: An Introduction. Washington USA, SPIE Press.
6. Gen, M., and Cheng, R. (1999). Genetic Algorithms and Engineering Optimization. United States: Wiley-Interscience.
7. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.523
Course Title: Computer Vision

Total Hours: 60

Course Objectives:
To help students review both the theoretical and practical aspects of computing with images for computer vision. To develop understanding in image formation, measurements, analysis and describe the geometric relationships between 2D images and the 3D world.

Course Outcomes
After completion of course, students would be able to:

- Describe the various image processing and analysis methods for computer vision.
- Compare and contrast various object and scene recognition, classification and clustering techniques.
- Develop the practical skills necessary to build computer vision applications.

UNIT I **14 Hours**
Overview, computer imaging systems, lenses, Image formation and sensing, Image analysis, pre-processing and binary image analysis.
Edge detection, Edge detection performance, Hough transform, corner detection
Activities: Assignment based and hands on training

UNIT II **16 Hours**
Segmentation, Morphological filtering, Fourier transform Feature extraction, shape, histogram, color, spectral, texture, using CVIP tools.
Activities: Learning through Experiment

UNIT III **16 Hours**
Feature analysis, feature vectors, distance /similarity measures, data pre-processing.
Pattern Analysis: Clustering: K-Means, K-Medoids, Mixture of Gaussians
Classification: Discriminant Function, Supervised, Un-supervised, Semi supervised
Activities: Review of various techniques

UNIT IV **14 Hours**
Classifiers: Bayes, KNN, ANN models; Dimensionality Reduction: PCA, LDA, ICA, and Non-parametric methods.
Recent trends in Activity Recognition, computational photography, Biometrics.
Activities: Case studies and group discussion

Transactional Modes:

- Lecture cum Demonstration
- Flipped Class Room
- E-tutorial

- Experimentation
- Online Teaching Tools

Suggested Readings:

1. Szeliski, R. (2011). Computer Vision - Algorithms and Applications. New York: Springer.
2. Goodfellow, I., Bengio Y., and Courville, A. (2017). Deep Learning. Cambridge: MIT Press.
3. Fisher, R. B., Dawson-Howe, K., and Fitzgibbon, A. (2013). Dictionary of Computer Vision and Image Processing, United States: Wiley.
4. Klette, R. (2014). Concise Computer Vision: An Introduction into Theory and Algorithms. New York: Springer.
5. Gose, E., Johnsonbaugh, R., and Steve. (2015). Pattern Recognition and Image Analysis. New Delhi: Pearson Education India.
6. Shinghal, R.. (2012). Pattern recognition: Techniques and applications. New Delhi: Oxford university press.
7. Bishop, C.M. (2012). Neural networks for pattern recognition. New Delhi: oxford university press.
8. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CBS.524
Course Title: Big Data Analytics and Visualization

Total Hours: 60

Course Objectives:
The course will help students prepare the big data for analytics and extract the meaningful data from unstructured big data. Help student to develop data visualizations skill and to apply various tools for analysis of structured and unstructured big data.

Course Outcomes
After completion of course, students would be able to:

- Illustrate the identification of Big Data problem
- Differentiate structured data from unstructured data.
- Use Hadoop related tools such as JAQL, Spark, Pig and Hive for structured and unstructured Big Data analytics

UNIT I **15 Hours**

Big Data Introduction: What is big data, why big data, convergence of key trends, unstructured data, industry examples of big data, web analytics, big data and marketing, fraud and big data, risk and big data, big data and healthcare, big data in medicine, advertising and big data, big data technologies, open source technologies, cloud and big data, mobile business intelligence, Crowd sourcing analytics, inter and trans firewall analytics.
Data Gathering and Preparation: Data formats, parsing and transformation, Scalability and real-time issues.
Activities: Case Study and Group Discussion

UNIT II **15 Hours**

Data Cleaning: Consistency checking, Heterogeneous and missing data, Data Transformation and segmentation.
Visualization: Descriptive and comparative statistics, Designing visualizations, Time series, Geo-located data, Correlations and connections, Hierarchies and networks, interactivity.
Activities: Implementation above theory with Python code

UNIT III **15 Hours**

Big Data Technology: Big Data Architecture, Big Data Warehouse, Functional Vs. Procedural Programming Models for Big Data
NoSQL: Introduction to NoSQL, aggregate data models, key-value and document data models.
Activities: Implementation and designing with Spark/MongoDB

UNIT IV **15 Hours**

Big Data Tools: Hadoop: Introduction to Hadoop Ecosystem, HDFS, Map-Reduce programming, Spark, PIG, JAQL, Understanding Text Analytics and Big Data, Predictive Analytics of Big Data, Role of Data Analytics.
Activities: Implementation and usage of tools over the cloud

Transactional Modes:

- Lecture cum Demonstration
- Peer Learning/Teaching
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. EMC Education Services. (2015). Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. United States: John Wiley & Sons.
2. Maheshwari, A. (2019). Data Analytics Make Accesible. California: Orilley Publications.
3. Croll, A., and Yoskovitz, B. (2013). Lean Analytics: Use Data to Build a Better Startup Faster. California: Oreilley Publications.
4. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CBS.523
Course Title: Secure Software Design

Total Hours: 60

Course Objectives:
To help students learn to fix software flaws and bugs in various software. To make students aware of various issues like weak random number generation, information leakage, poor usability, and weak or no encryption on data traffic.
Expose students to techniques for successfully implementing and supporting network services on an enterprise scale and heterogeneous systems environment.

Course Outcomes
After completion of course, students would be able to:

- Show Interrelationship between security and software development process.
- Differentiate between various software vulnerabilities.
- Explain software process vulnerabilities for an organization.
- Recognize resources consumption in a software.

UNIT I **13 Hours**

Secure Software Design
Identify software vulnerabilities and perform software security analysis, Master security programming practices, Master fundamental software security design concepts, perform security testing and quality assurance.
Activities: Case study based learning

UNIT II **17 Hours**

Enterprise Application Development
Describe the nature and scope of enterprise software applications, Design distributed N-tier software application, Research technologies available for the presentation, business and data tiers of an enterprise software application, Design and build a database using an enterprise database system, develop components at the different tiers in an enterprise system, Design and develop a multi-tier solution to a problem using technologies used in enterprise system, Present software solution.
Activities: Group Discussion based learning

UNIT III **15 Hours**

Enterprise Systems Administration
Design, implement and maintain a directory-based server infrastructure in a heterogeneous systems environment, Monitor server resource utilization for system reliability and availability, Install and administer network services (DNS/DHCP/Terminal Services/Clustering/Web/Email).
Activities: Group discussion based learning

UNIT IV **15 Hours**

Obtain the ability to manage and troubleshoot a network running multiple services, Understand the requirements of an enterprise network and how to go about managing them.

Handle insecure exceptions and command/SQL injection, Defend web and mobile applications against attackers, software containing minimum vulnerabilities and flaws. Case study of DNS server, DHCP configuration and SQL injection attack.

Activities: Case study of Various server configuration

Transactional Modes:

- Lecture cum Demonstration
- Peer Learning/Teaching
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Richardson, T., and Thies, C. N. (2012). Secure Software Design. Massachusetts: Jones & Bartlett Learning.
2. Kenneth, R. Van, W., Mark, G., Graff, D.S., Peters, D. L., Burley, Enterprise Software Security: A Confluence of Disciplines, United States: Addison -Wesley, Professional.
3. McGraw, G. (2006). Software Security: Building Security. New Delhi: Tata McGraw.
4. Stuttard, D. (2011). The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws. United States: Wiley.
5. Solem, J. E. (2012). Programming Computer Vision with Python: Tools and algorithms for analyzing images. California: O'Reilly Media.
6. Fernandez, E. B. (2013). Designing secure architecture using software patterns, United Kingdom: John Wiley & sons limited.
7. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.524
Course Title: Internet of Things

Total Hours: 60

Course Objectives:
The objective of this course is to introduce the students to the concepts of IoT, its networking and communication. The course focussed on use of IoT technology and its design constraints.

Course Outcomes
After completion of course, students would be able to:
- Describe IOT and its networking and communication aspects.
- Analyze the challenges in IoT Design
- Design IoT applications on different embedded platform.

UNIT I **15 Hours**
Introduction to IoT: Defining IoT, Characteristics of IoT, Physical design of IoT, Logical design of IoT, Functional blocks of IoT, Communication models and APIs IoT and M2M, Difference between IoT and M2M, Software define Network.
Activities: Group Discussion and Case studies

UNIT II **15 Hours**
Network and Communication aspects: Wireless medium access issues, MAC protocol survey, Survey routing protocols, Sensor deployment, Node discovery, Data aggregation and Dissemination.
Activities: Flip Learning with simulation tools

UNIT III **15 Hours**
Challenges in IoT Design: challenges, Development challenges, Security challenges, Other Challenges
Domain specific applications: IoT Home automation, Industry applications, Surveillance applications, Other IoT applications
Activities: Group Discussion and IOT design simulation using simulation tools

UNIT IV **15 Hours**
Developing IoTs: Developing applications through IoT tools including Python/Arduino/Raspberry pi, Developing sensor based application through embedded system platform.
Activities: Hands on experience with IOT kits

Transactional Modes:
- Lecture cum Demonstration
- Collaborative
- Experimentation
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Madisetti, V., & Bahga, A. (2015). Internet of Things: A Hands-On Approach, New Delhi: Orient Blackswan Pvt. Ltd.
2. Dargie, W., & Poellabauer, C. (2010). Fundamentals of Wireless Sensor Networks: Theory and Practice. United States: Wiley-Blackwel.
3. DaCosta, F., & Henderson B. (2014). Rethinking the Internet of Things: A Scalable Approach to Connecting Everything, New York: Apress Publications.
4. Holler, J., Tsiatsis V., Mulligan, C., Avesand, S., Karnouskos, S., & Boyle, D. (2014). From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence. Massachusetts: Academic Press.
5. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CBS.525
Course Title: Secure Coding

Total Hours: 60

Course Objectives:
The outcome of this course is to explain the most frequent programming errors leading to software vulnerabilities and identify security problems in software.

Course Outcomes
After completion of course, students would be able to:

- Define secure programs and list various risks in the softwares.
- Classify different errors that lead to vulnerabilities.
- Analyze various possible security attacks.

UNIT I **15 Hours**

Software Security: Security Concepts, Security Policy, Security Flaws, Vulnerabilities, Exploitation and Mitigations. Software Security problems, Classification of Vulnerabilities.
Security Analysis: Problem Solving with static analysis: Type Checking, Style Checking, Program understanding, verifications and property checking, Bug finding and Security Review.
Activities: Group Discussion based learning

UNIT II **15 Hours**

Strings: Common String manipulating Errors, String Vulnerabilities and Exploits, Mitigation Strategies for strings, String handling functions, Runtime protecting strategies, Notable Vulnerabilities.
Integer Security: Integer data Type, Integer Conversions, Integer Operations, Integer Vulnerabilities, Mitigation Strategies.
Activities: Implementation of above concepts in various programming Languages

UNIT III **15 Hours**

Handling Inputs: What to validate, How to validate, Preventing metadata Vulnerabilities.
Buffer Overflow: Introduction, Exploiting buffer overflow vulnerabilities, Buffer allocation strategies**,** Tracking buffer sizes, buffer overflow in strings, Buffer overflow in Integers Runtime protections
Activities: Implementation of above concepts in various programming Languages

UNIT IV **15 Hours**

Web Applications: Input and Output Validation for the Web: Expect That the Browser Has Been Subverted, HTTP Considerations: Use POST, Not GET, Request Ordering, Error Handling, Request Provenance
Maintaining Session State: Use Strong Session Identifiers, Enforce a Session Idle Timeout and a Maximum Session Lifetime, Begin a New Session upon Authentication.

Activities: Implementation of above concepts in various programming Languages

Transactional Modes:

- Lecture
- Case Studies
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Seacord, R. C. (2013). Secure Coding in C and C++. United States: Addison Wisley Professional.
2. Chess, B., and West J. (2007). Secure Programming with static Analysis. United States: Addison Wisley.
3. Seacord, R. C. (2009). The CERT C Secure Coding Standard. Pearson Education, United States: Addison-Wesley.
4. Howard, M., LeBlanc, D. (2002). Writing Secure Code. New Delhi: Pearson Education.
5. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.525
Course Title: GPU Computing

Total Hours: 60

Course Objectives:
To help students learn parallel programming with Graphics Processing Units (GPUs).

Course Outcomes
After completion of course, students would be able to:

- Explain parallel programming
- Demonstrate programing on GPUs
- Outline the process of debugging and profiling parallel programs.
- Design various complex problems using GPU computing

UNIT I **15 Hours**

Introduction: History, Graphics Processors, Graphics Processing Units, GPGPUs. Clock speeds, CPU / GPU comparisons, Heterogeneity, Accelerators, Parallel programming, CUDA Open CL / Open ACC, Hello World Computation Kernels, Launch parameters, Thread hierarchy, Warps / Wave fronts, Thread blocks / Workgroups, Streaming multiprocessors, 1D / 2D /3D thread mapping, Device properties, Simple Programs.
Activities: Assignment Based Learning, Case studies, Simple Implementation

UNIT II **15 Hours**

Memory: Memory hierarchy, DRAM / global, local / shared, private / local, textures, Constant Memory, Pointers, Parameter Passing, Arrays and dynamic Memory, Multi-dimensional Arrays, Memory Allocation, Memory copying across devices, Programs with matrices, Performance evaluation with different memories.
Activities: Implementation Based Learning, Matrices Computation.

UNIT III **14 Hours**

Synchronization: Memory Consistency, Barriers (local versus global), Atomics, Memory fence. Prefix sum, Reduction. Programs for concurrent Data Structures such as Worklists, Linked-lists. Synchronization across CPU and GPU
Functions: Device functions, Host functions, Kernels functions, Using libraries (such as Thrust), and developing libraries.
Activities: Assignment Based Learning, Implementation of Worklists and Linklists, Live Demonstration

UNIT IV **16 Hours**

Support: Debugging GPU Programs. Profiling, Profile tools, Performance aspects.
Streams: Asynchronous processing, tasks, Task-dependence, Overlapped data transfers, Default Stream, Synchronization with streams. Events, Event-based-Synchronization - Overlapping data transfer and kernel execution, pitfalls.
Case Studies: Image Processing, Graph algorithms, Simulations, Deep Learning

Advanced topics: Dynamic parallelism, Unified Virtual Memory, Multi-GPU processing, Peer access, Heterogeneous processing.
Activities: Case studies, Live implementation

Transactional Modes:

- Lecture
- Flipped Class
- E-tutorial
- Programme Learning
- Online Teaching Tools

Suggested Readings:

1. Kirk, D., Hwu, W., and Kaufman, M. (2010). Programming Massively Parallel Processors: A Hands-on Approach. Massachusetts: Morgan Kaufmann.
2. Cook, S., and Kaufman, M. (2014). CUDA Programming: A Developer's Guide to Parallel Computing with GPUs. Elsevier.
3. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.529
Course Title: Blockchain Technology

Total Hours: 60

Course Objectives:
The outcome of this course is to introduce students to the concept of Blockchain, crypto primitives, Bitcoin basics, distributed consensus, consensus in Bitcoin, permissioned Blockchain, hyper ledger fabric and various applications where Blockchain is used.

Course Outcomes
After completion of course, students would be able to:

- Describe the basic concept of Blockchain, Crypto Primitives, Bitcoin Basics
- Identify the area in which they can apply permission or permission less blockchain.
- Apply Block chaining concept in various applications.

UNIT I **15 Hours**

Introduction to Blockchain: What is Blockchain, Public Ledgers, Blockchain as Public Ledgers, Bitcoin, Blockchain 2.0, Smart Contracts, Block in a Blockchain, Transactions, Distributed Consensus, The Chain and the Longest Chain, Cryptocurrency to Blockchain 2.0, Permissioned Model of Blockchain
Activities: Case studies based Learning, Group Discussion.

UNIT II **15 Hours**

Basic Crypto Primitives: Cryptographic Hash Function, Properties of a hash function, Hash pointer and Merkle tree, Digital Signature, Public Key Cryptography, A basic cryptocurrency.
Bitcoin Basics: Creation of coins, Payments and double spending, FORTH – the precursor for Bitcoin scripting, Bitcoin Scripts, Bitcoin P2P Network, Transaction in Bitcoin Network, Block Mining, Block propagation and block relay.
Activities: Live Demonstration , Implementation Based Learning of hash functions, Group Discussions

UNIT III **15 Hours**

Distributed Consensus: Why Consensus, Distributed consensus in open environments, Consensus in a Bitcoin network.
Consensus in Bitcoin: Bitcoin Consensus, Proof of Work (PoW) – basic introduction, Hashcash PoW, Bitcoin PoW, Attacks on PoW and the monopoly problem, Proof of Stake, Proof of Burn and Proof of Elapsed Time. The life of a Bitcoin Miner, Mining Difficulty, Mining Pool.

Permissioned Blockchain: Permissioned model and use cases, Design issues for Permissioned blockchains, Execute contracts, State machine replication, Consensus models for permissioned blockchain, Distributed consensus in closed environment, Paxos, RAFT Consensus, Byzantine general problem.

Activities: Group Discussion, Assignment Based Learning, Case studies

UNIT IV **15 Hours**

Blockchain Components and Concepts: Actors in a Blockchain, Components in Blockchain design, Ledger in Blockchain.

Hyperledger Fabric – Transaction Flow: Fabric Architecture, Transaction flow in Fabric.

Hyperledger Fabric Details: Ordering Services, Channels in Fabric, Fabric Peer and Certificate Authority.

Fabric – Membership and Identity Management: Organization and Consortium Network, Membership Service Provide, Transaction Signing.

Activities: Assignment Based Learning, Live Demonstration.

Transactional Modes:

- Lecture cum Demonstration
- Blended Learning
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Gaur, N., Desrosiers, L., Ramakrishna, V., Novotny, P., Baset, S., and O'Dowd A. (2018). Hands-On Blockchain with Hyperledger: Building decentralized applications with Hyperledger Fabric and Composer. United Kingdom: Packt Publishing Ltd. Packt.
2. Badr, B., Horrocks, R., and Xun(Brian), Wu. (2018). Blockchain By Example: A developer's guide to creating decentralized applications using Bitcoin, Ethereum, and Hyperledger. United Kingdom: Packt Publishing Ltd.
3. Dhillon, V., Metcalf D., and Hooper M. (2017). Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You.New York: Apress.
4. Mukhopadhyay M. (2018). Ethereum Smart Contract Development: Build blockchain-based decentralized applications using solidity. United States: Packt Publishing Ltd.
5. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CBS.527
Course Title: Digital Forensics

Total Hours: 60

Course Objectives:
The course provides an in-depth study of the rapidly changing and fascinating field of computer forensics. Introduces the students to the technical expertise and the knowledge required to investigate, detect and prevent digital crimes.

Course Outcomes
After completion of course, students would be able to:

- Describe relevant legislation and codes of ethics.
- Explain computer forensics, digital detective and various processes, policies and procedures.
- Apply E-discovery, guidelines and standards, E-evidence, tools and environment.
- Analyse Email and web forensics and network forensics.

UNIT I **15 Hours**

Digital Forensics Science: Forensics science, computer forensics, and digital forensics.
Computer Crime: Criminalistics as it relates to the investigative process, analysis of cyber-criminalistics area, holistic approach to cyber-forensics.
Legal Aspects of Digital Forensics: IT Act 2000, amendment of IT Act 2008.
Activities: Analysis of Cyber Attacks and laws with case studies

UNIT II **15 Hours**

Incident- Response Methodology, Cyber Crime Scene Analysis: Discuss the various court orders etc., methods to search and seizure electronic evidence, retrieved and un-retrieved communications, Discuss the importance of understanding what court documents would be required for a criminal investigation.
Activities: Preparation of various documents related to Cyber Crime Investigation.

UNIT III **15 Hours**

Image Capturing, Authenticating Evidence, Hidden Data Extraction, Data Storage, File Systems, Recovery of deleted files, Cracking Passwords,
Internet Crime Investigations, Web Attack Investigations.
Activities: Demonstration of various tools to perform digital forensics

UNIT IV **15 Hours**

Computer Forensics: Prepare a case, Begin an investigation, Understand computer forensics workstations and software, Conduct an investigation, Complete a case, Critique a case.
Network Forensics: open-source security tools for network forensic analysis, requirements for preservation of network data.
Mobile Forensics: mobile forensics techniques, mobile forensics tools.

Activities: Analysis of Case Studies, Performing various activities to perform network and mobile forensics.

Transactional Modes:

- Lecture
- Case Studies
- Collaborative
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Sammons, J. (2014). The Basics of Digital Forensics, Elsevier.
2. Davidoff, S., and Ham, J. (2012). Network Forensics Tracking Hackers through Cyberspace. United States: Prentice Hall.
3. Solomon, M. G., Rudolph, K., Tittel, E., Broom, N., and Barrett, D. (2011). Computer Forensics Jump Start. United States: Willey Publishing.
4. Marcella, A. J., Cyber forensics: A field manual for collecting, examining and preserving evidence of computer crimes. New York: Auerbach publications.
5. Davidoff, S. (2012). Network forensics: Tracking hackers through cyberspace. New Delhi: Pearson education India.
6. Godbole, Nina, Belapure, Sunit (2011). Cyber security: Understanding cyber crimes, computer forensics and legal perspectives. New Delhi: Wiley India.
7. Casey, Eoghan (Ed.). (2010). Handbook of digital forensics and investigation, Amsterdam,: Academic Press.
8. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CBS.530
Course Title: Quantum Computing & Cryptography

Total Hours: 60

Course Objectives:

- To provide fundamental concepts of quantum information processing and cryptography, and take the discussion forward to potentials offered, technological bottlenecks and the way forward.
- To expose the participants to the state-of-the-art in quantum computing and cryptography with its possible impact on the society.

Course Outcomes

- Participants will understand the basic concepts and terminologies in quantum information processing and quantum cryptography.
- To work in the field of quantum information processing and quantum cryptography, and to design efficient quantum algorithms to solve different computing problems.
- To design new or modify existing quantum cryptographic algorithms for secure key distribution and communications.
- To grasp the working principle of a quantum computer and understand the impact of noise in real world implementations.
- To understand some of the long-standing issues in quantum computing, and way forward in Noise-Intermediate-Scale-Quantum and Post Quantum Cryptography era.
- To understand the current scenario in Google, IBM, D-wave, IonQ etc.

Unit I
Basics of Quantum Information and Linear Algebra: Why Quantum Computing, Classical to quantum mechanics, Hilber space, bases and linear independence, operators and matrices, Hermitian and Unitary operators, measurements in quantum mechanics, Einstein-Podolsky-Rosen paradox
Activities: Exercise based learning, Demonstration of above theory using Mathemetica/ MATLAB tools

Unit II
Introduction to quantum information: Qubits and quantum gates, quantum circuits, density operators, pure and mixed states, Bloch sphere, Bell states, information and entropy, von-Neumann entropy and trace distance, fidelity, No-cloning Theorem

Activities: Assignment based learning, Demonstration of above theory using Mathemetica/ MATLAB tools

Unit III

Entanglement and Nonlocality: Quantum entanglement, bi-partite and multiqubit systems, Bell-type inequalities and nonlocality, entanglement classes and measures, quantum parallelism, Deutsch-Jozsa algorithm.

Activities: Assignment based learning, Demonstration of Entanglement and Non-locality through animated videos.

Unit IV

Applications and Quantum Cryptography: Teleportation, dense coding, entanglement swapping, quantum key distribution, quantum cryptographic protocols.

Quantum Noise and Operation: Environments and quantum operations, examples of noisy channels, effect of noise on entanglement and efficiency of communication protocols.

Activities: Demonstration of above theory using Mathemetica/ MATLAB tools, Case based study of realization of quantum computing.

Transactional Modes:

- Lecture cum Demonstration
- Peer Learning/Teaching
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Nielsen, M. A. and Chuang, I. L., (2010), Quantum Computation and Quantum Information, 10th Anniversary addition, Cambridge University Press
2. Griffiths, D. J., (2016), Introduction to Quantum Mechanics, Reprint edition, Pearson Prentice Hall, 2006.
3. Bouwmeester, D., Ekert, A. and Zeilinger, A., (2000), The Physics of Quantum Information, Reprint edition, Springer Berlin Heidelberg.
4. Research Articles
5. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.508
Course Title: Machine Learning

Total Hours: 60

Course Objectives:
To help students explain the concept of how to learn patterns and concepts from data without being explicitly programmed. To analyze various machine learning algorithms and techniques with a modern outlook focusing on recent advances.

Course Outcomes
After completion of course, students would be able to:

- Describe machine learning approaches.
- Discuss features that can be used for a particular machine learning approach in various applications.
- Compare and contrast pros and cons of various machine learning techniques.
- To mathematically analyze various machine learning approaches and paradigms.
- Formulate various machine learning and ensemble methods for use in IOT applications.

UNIT I **16 Hours**

Introduction to learning Techniques: Supervised Learning (Regression/Classification)

- Basic methods: Distance-based methods, Nearest-Neighbours, Decision Trees, Naive Bayes
- Linear models: Linear Regression, Logistic Regression, Generalized Linear Models
- Support Vector Machines, Nonlinearity and Kernel Methods
- Beyond Binary Classification: Multi-class/Structured Outputs, Ranking

Activities: Brainstorming, assignment based learning

UNIT II **14 Hours**

Unsupervised Learning

- Clustering: K-means/Kernel K-means
- Dimensionality Reduction: PCA and kernel PCA
- Matrix Factorization and Matrix Completion
- Generative Models (mixture models and latent factor models)

Activities: Exercise based learning and practical hands on training

UNIT III **14 Hours**

Evaluating Machine Learning algorithms and Model Selection, Introduction to Statistical Learning Theory, Ensemble Methods (Boosting, Bagging, Random Forests).

Sparse Modeling and Estimation, Modeling Sequence/Time-Series Data, Deep Learning and Feature Representation Learning.

Introduction to ANN and Deep learning.

Activities: Exercise based learning and practical hands on training

UNIT IV **16 Hours**

Scalable Machine Learning (Online and Distributed Learning) A selection from some other advanced topics, e.g., Semi-supervised Learning, Active Learning, Reinforcement Learning, Inference in Graphical Models, Introduction to Bayesian Learning and Inference.

Simulation Tool for Machine Learning, Hands on with recent tools WEKA, R, MATLAB

Recent trends in various learning techniques of machine learning and classification methods for IOT applications. Various models for IOT applications.

Activities: Analysis of various case studies

Transactional Modes:

- Lecture cum Demonstration
- Collaborative Learning
- Peer Learning/Teaching
- Experimentation
- Online Teaching Tools

Suggested Readings:

1. Murphy, K. (2012). Machine Learning: A Probabilistic Perspective. Cambridge: MIT Press.
2. Hastie, T., Tibshirani, R., and Friedman, J. (2017). The Elements of Statistical Learning. New York: Springer.
3. Bishop, C. (2011). Pattern Recognition and Machine Learning. New York: Springer.
4. Shalev-Shwartz, S., and Ben-David, S. (2014). Understanding Machine Learning: From Theory to Algorithms. New Delhi: Cambridge University Press.
5. Kubat, M. (2015). An introduction to machine learning, New York: Springer Science.
6. Barber, D. (2014). Bayesian reasoning and machine learning. New Delhi: Cambridge University Press.
7. Flach, P. (2015). Machine Learning. New Delhi: Cambridge University Press.
8. Mitchell, M.T. (2013). Machine Learning. New Delhi: Tata McGraw Hill Education Private Limited.
9. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
0	0	4	2

Course Code: CST.527
Course Title: Soft Computing Lab
Course Objectives: The primary outcomes of this lab course is to provide a practical introduction to various techniques in soft computing and their applications.

Learning outcome:
After Completion of the lab course the students will be able to:

- Create programs to implement simple applications using the fuzzy logic.
- Distinguish various types of neural networks and write programmes to implement the same.
- Use optimization based on GA and implement some of its applications.

Lab Assignments

1. Implement perceptron and show its working on NAND gate.
2. Implement multilayer perceptron for XOR gate
3. Write a program to implement a Backpropagation neural network from scratch. Then use it to implement a parity checker.
4. Write a program to implement ART1 and use it to learn Alphabets.
5. Implement various membership functions for fuzzifying the crisp values.
6. Implement various defuzzification methods
7. Develop a fuzzy inference system for modelling the tip given to e-commerce delivery boys based on the customer feedback.
8. Implement various techniques for selection, crossover and mutation in Genetic Algorithms.
9. Implement a simple the genetic application.
10. Implement a simple neurofuzzy system

Lab Evaluation:
The evaluation of lab criteria will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

Suggested Readings:

1. Lab Manual
2. Kumar, U.D., and Pradhan, M. (2019). Machine Learning using Python. Wiley.

L	T	P	Cr
0	0	2	1

Course Code: CST.533
Course Title: Computer Vision Lab

Course Objectives:
The objectives of the Computer Vision Lab course are to introduce students to the basic concepts and techniques of Computer Vision. To develop skills of using recent Computer Vision software for solving practical problems.

Course Objectives:
After completion of course, students would be able to:

- Implement edge detection and segmentation algorithms.
- Apply common feature extraction algorithms in practice and implementing the same.
- Perform experiments in Computer Vision using real-world data for Pattern Analysis.

Suggested Readings:
Lab Manual

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

L	T	P	Cr
0	0	2	1

Course Code: CBS.534
Course Title: Big Data Analytics and Visualization Lab

Course Objectives:
The lab will help students prepare the big data with pre-procesing analysis and to extract the meaningful data from unstructured data. Help student to develop data visualizations skill and to apply various tools for analysis of structured and unstructured big data.

Course Outcomes
After completion of lab course, students would be able to:
- Pre-process the un-structured data by various cleaning activities.
- Convert the un-structured data to structured format.
- Use Python libraries for analysis and visualisation of data such as PySpark, PyMongo,pandas, numpy and beutifulsoap.

Suggested Readings:
Lab Manual

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	Marks
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

L	T	P	Cr
0	0	2	1

Course Code: CBS.539
Course Title: Secure Software Design Lab

Course Objectives:
To fix software flaws and bugs in various software.Students will aware of various issues like weak random number generation, information leakage, poor usability, and weak or no encryption on data traffic. Learn Methodologies and tools for developing secure software with minimum vulnerabilities and flaws.

Course Outcomes
After completion of course, students would be able to:
- Learn the use of various tools for software vulnerability.
- Apply different techniques for identification of software flaws.
- Track the resolution of flaws in software.
- Interrelate security and software development process.

Suggested Readings:
Lab Manual

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	Marks
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

L	T	P	Cr
0	0	2	1

Course Code: CST.534
Course Title: IOT (Internet of Things) Lab

Course Objectives:
The objective of IOT Lab is to introduce the students to the different IOT technologies. To develop skills that will help the students to develop different IOT applications. To help use different IOT protocols and analysis the data in IOT.

Course Outcomes
After completion of course, students would be able to:

- Identify the different technology and develop IOT based applications.
- Analysis and evaluate protocols used in IOT.
- Evaluate the data received through sensors in IOT**.**

Suggested Readings:
Lab Manual

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

L	T	P	Cr
0	0	2	1

Course Code: CBS.536
Course Title: Secure Coding Lab

Course Objectives:
The objective of this course is to explain the most frequent programming errors leading to software vulnerabilities and identify security problems in software.

Course Outcomes
- Implement secure programs and list various risks in the softwares.
- Classify different errors that lead to vulnerabilities.
- Analyse various possible security attacks in the programs.

Suggested Readings:

1. Lab Manual
2. 1. Seacord, R. C. (2013). Secure Coding in C and C++. United States: Addison Wisley Professional.
3. Chess, B., and West J. (2007). Secure Programming with static Analysis. United States: Addison Wisley

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

L	T	P	Cr
0	0	2	1

Course Code: CST.535
Course Title: GPU Computing Lab

Course Objectives:
The objective of GPU Computing is to introduce the fundamentals of graphics processing units and corresponding programming environments. Introduce the learner to fundamental and advanced parallel algorithms through the GPU programming environments.

Course Outcomes
After completion of course, students would be able to

- To design, formulate, solve and implement high performance versions of standard single threaded algorithms.
- To demonstrate the architectural features in the GPU hardware accelerators.
- To design and deploy parallel programs.

Suggested Readings:
Lab Manual

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

L	T	P	Cr
0	0	2	1

Course Code: CST.536
Course Title: Blockchain Technology Lab

Course Objectives:
The objective of this course is to introduce students to the concept of Blockchain, crypto primitives, Bitcoin basics, distributed consensus, consensus in Bitcoin, permissioned Blockchain, hyper ledger fabric and various applications where Blockchain is used.

Course Outcomes
- Design the basic concept of Blockchain, Crypto Primitives, Bitcoin Basics
- Identify the area in which they can apply permission or permission less blockchain.
- Apply Block chaining concept in various applications.

Suggested Readings:
1. Lab Manual
2. Gaur, N., Desrosiers, L., Ramakrishna, V., Novotny, P., Baset, S., & O'Dowd A. (2018). Hands-On Blockchain with Hyperledger: Building decentralized applications with Hyperledger Fabric and Composer. United Kingdom: Packt Publishing Ltd. Packt.
3. Badr, B., Horrocks, R., and Xun(Brian), Wu. (2018). Blockchain By Example: A developer's guide to creating decentralized applications using Bitcoin, Ethereum, and Hyperledger. United Kingdom: Packt Publishing Ltd.

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

L	T	P	Cr
0	0	2	1

Course Code: CBS.535
Course Title: Digital Forensics Lab

Course Objectives:

The objective of this course is to provide practical exposure of tools used to perform various activities related to different types of digital forensics such as memory forensics, network forensics and web forensics.

Course Outcomes

After completion of this lab course, students would be able to:
- Prepare case documents.
- Setup platform for digital investigation.
- Acquire and analyse various types of electronic evidences..
- Analyse Email and web communication headers.

Suggested Readings:

1. Lab Manual
2. Marcella, A. J.(2007), Cyber forensics: A field manual for collecting, examining and preserving evidence of computer crimes. New York: Auerbach publications.

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	Marks
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

L	T	P	Cr
0	0	2	1

Course Code: CBS.538
Course Title: Quantum Computing & Cryptography Lab

Course Objectives:

- To provide one-to-one correspondence between theory and hands-on in terms of in-depth knowledge of fundamentals of Quantum Information Processing.
- To develop skills with hand-on experience of simulation of quantum computation in order to work in the field of Quantum Information Processing and Cryptography.
- To acquire deeper understanding to design, develop, and analyse efficient algorithms in the field of Quantum Computing.

Course Outcomes:

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

- Write a script to simulate qubits, multi-qubit pure and mixed quantum states, the celebrated Bell states and density matrices associated with entangled systems.
- Write a script to simulate quantum circuits composed of single and multi-qubit quantum gates.
- Write a script to simulate different measures of entanglement and nonlocality in pure and mixed two and three-qubit states.
- Write a script to simulate different noisy channels to analyse the effect of noise on entanglement and efficiency of a protocol.
- Simulate different quantum information processing protocols such as teleportation, dense coding, and Secret Sharing.

Students will implement the lab practical as per the syllabus of the subject.

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

Suggested Readings

L	T	P	Cr
0	0	2	1

Course Code: CST.517
Course Title: Machine Learning Lab

Course Objectives:
The objectives of the Machine Learning Lab course are to introduce students to the basic concepts and techniques of Machine Learning. To develop skills of using recent machine learning software for solving practical problems.

Course Outcomes
After completion of course, students would be able to:

- Review some common Machine Learning algorithms and their limitations.
- Apply common Machine Learning algorithms in practice and implementing the same.
- Perform experiments in Machine Learning using real-world data.

Suggested Readings:

1. Lab Manual
2. Kumar, U.D., and Pradhan, M. (2019). Machine Learning using Python. Wiley.

List of Practical will be based on Elective subject opted by the students

Lab Evaluation:
The criteria for evaluation of lab will be based on following parameters:

Component	**Marks**
Continuous Evaluation	60
End Term (Implementation and Viva-Voce)	40
Total	**100**

Value Added Course

As per the availability of faculty

L	T	P	Cr
0	0	2	1

Course Code: CBS.504
Course Title: Report writing using LaTeX

Total Hours: 32

Course Outcomes

After the completion of course, participants will be able to:

- Use the basic commands in Latex.
- Develop scripts in Latex for different type of documents.
- Illustrate troubleshooting in the latex scripts.

UNIT I **8 Hours**

Latex Introduction: Installing and setting Latex environment in Windows and Linux.

Document Structure: Essential in preparing the structure of documents, Creating Titles at different levels, Sections, Labelling and preparing Table of Contents.

Activities: Live Demonstration of LaTeX scripts.
Assignment to write the LaTeX scripts.

UNIT II **8 Hours**

Formatting Text: Font Effects, Colored Text, Font Size, Bullets and lists, Comments, Spacing and Special Characters.

Activities: Live Demonstration of LaTeX scripts.
Assignment to write the LaTeX scripts.

UNIT III **8 Hours**

Tables: Working with tables, Styles, Borders, Wrapping, Inserting new rows columns and caption of Tables.

Figures: Working with Figures, Formatting of Figures, caption, Alignment and wrapping Text around figures.

Activities: Live Demonstration of LaTeX scripts.
Assignment to write the LaTeX scripts.

UNIT IV **8 Hours**

Equation: Inserting Equation, Mathematical Symbols, Fractions, Roots, Sums & Integrals and Greek Letters.

References: BibTeX File, Inserting the bibliography, Citing References, Styles of References

Activities: Live Demonstration of LaTeX scripts.
Assignment to write the LaTeX scripts.

Transactional Modes:

- Lecture
- Peer Learning/Teaching
- E-tutorial
- Self-Learning

- Online Teaching Tools

Suggested Readings:

1. Lamport, L. (2014), Latex A document preparation system. New York: Adisson Wesley Publishing Company.
2. Kotwiz. S. (2015). Latex Cook Book. United Kingdom: Packt Publishing Lmt.
3. Nicola Louise Cecilia Talbot. (2013). Using LaTeX to Write a PhD Thesis, Dickimaw Books.
4. Research Articles from SCI & Scopus indexed Journals.

Value Added Course
(For other departments only as per the availability of faculty)

L	T	P	Cr
0	0	2	1

Course Code: CST.504
Course Title: Python Programming

Total Hours: 32

Course Outcomes
After the completion of course, participants will be able to:

- Explain basics of programming.
- Define various constructs of python programming.
- Develop python code to handle data stored in files.
- Develop python code to represent the data in graphical mode.

UNIT I **8 Hours**
Introduction to algorithm, flowchart and programming,
Python Introduction, Installing and setting Python environment, variables and its types, Operators. Flow control: if, if-else, for, while, range() function, continue statement, pass statement.
Activities: Lab based practices for above concepts

UNIT II **8 Hours**
Lists: Basic Operations, Iteration, Indexing, Slicing. Dictionaries: Basic dictionary operations, Basic String operations
Activities: Lab based practices for above concepts

UNIT III **8 Hours**
Functions: Definition, Call, Arguments. Pattern Matching with Regular Expressions, Introduction to pandas library, plotting data using matplotlib
Activities: Lab based practices for above concepts

UNIT IV **8 Hours**
File handling: Reading and Writing Files, working with Excel Spreadsheets, working with PDF and Word Documents, working with CSV Files
Activities: Lab based practices for above concepts

Transactional Modes:

- Lecture
- Blended Learning
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Sweigart, AI. (2014). Automate the Boring Stuff with Python Practical Programming for Total Beginners. Switzerland: No Starch Press.
2. Mark, L. (2013). Learning Python. California: Oreilly Media.
3. Research Articles from SCI & Scopus indexed Journal.

SEMESTER-III

L	T	P	Cr
4	0	0	4

Course Code: CST.551 `

Course Title: Optimization Techniques

Total Hours: 60

Course Objectives:

The outcome of this course is to provide insights to the mathematical formulation of real world problems and to optimize these mathematical problems using nature based algorithms. And the solution is useful especially for NP-Hard problems.

Course Outcomes

After completion of course, students would be able to:

- Formulate optimization problems.
- Explain and apply the concept of optimality criteria for various types of optimization problems.
- Solve various constrained and unconstrained problems in Single variable as well as multivariable.
- Apply the methods of optimization in real life situations.

UNIT I **15 Hours**

Engineering application of Optimization, Formulation of design problems as mathematical programming problems.

General Structure of Optimization Algorithms, Constraints, the Feasible Region.

Activities: Assignment Based Learning, Practice Examples

UNIT II **15 Hours**

Branches of Mathematical Programming: Optimization using calculus, Graphical Optimization, Linear Programming, Quadratic Programming, Integer Programming, Semi Definite Programming.

Activities: Numerical Based Questions

UNIT III **15 Hours**

Optimization Algorithms like Genetic Optimization, Particle Swarm Optimization, Ant Colony Optimization etc.

Activities: Implementation of algorithms, Numerical Based questions for Genetic optimization

UNIT IV **15 Hours**

Real life Problems and their mathematical formulation as standard programming problems. Recent trends: Applications of ant colony optimization, genetics and linear and quadratic programming in real world applications.

Activities: Case studies, Group Discussions, Presentations by students.

Transactional Modes:

- Lecture cum Demonstration
- Peer Learning/Teaching

- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Wolsey, L. (1998). Integer programming. United States: Wiley-Interscience.
2. Antoniou, A., and Wu-Sheng, Lu. (2007). Practical Optimization Algorithms and Engineering Applications. New Delhi: Springer.
3. Edwin, K., Chong, P., and Zak S. H. (2017). An Introduction to Optimization, New Delhi: Wiley-India.
4. Bertsimas, D., & Weismantel, R. (2005). Optimization over integers. Waltham: Dynamic Ideas.
5. Karlof, J. K. (2005). Integer programming: theory and practice. London: CRC Press Inc.
6. Williams, H. P. (2010). Logic and Integer Programming. New York: Springer.
7. Chen, D., Batson, R. G., and Dang, Y., (2010). Applied Integer Programming: Modelling and Solution. United States: John Wiley and Sons.
8. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.552
Course Title: Data Warehousing and Data Mining

Total Hours: 60

Course Objectives:
The outcome of this course is to introduce data warehousing and mining techniques. Applications of data mining in web mining, pattern matching and cluster analysis are included to aware students of broad data mining areas.

Course Outcomes
After completion of course, students would be able to:

- Discuss different sequential pattern algorithms.
- Apply the techniques to extract patterns from time series data and their applications in real world.
- Examine Graph mining algorithms to Web mining.
  Design the computing framework for Big Data.

UNIT I **14 Hours**

Introduction to Data Warehousing: Data warehousing Architecture, OLAP Server, Data warehouse Implementation.
Data Mining: Mining frequent patterns, association and correlations; Sequential Pattern Mining concepts, primitives, scalable methods;
Activities: Brainstorming for finding the Association rules, Case study to illustrate the data warehouse and data mining model design principles.

UNIT II **15 Hours**

Classification and prediction: Cluster Analysis – Types of Data in Cluster Analysis, Partitioning methods, Hierarchical Methods; Transactional Patterns and other temporal based frequent patterns.
Activities: Assignment based learning, Exercise based learning.

UNIT III **16 Hours**

Mining Time series Data, Periodicity Analysis for time related sequence data, Trend analysis, Similarity search in Time-series analysis;
Mining Data Streams, Methodologies for stream data processing and stream data systems, Frequent pattern mining in stream data, Sequential Pattern Mining in Data Streams, Classification of dynamic data streams.
Activities: Case based study and Group discussion for the prediction of solutions for real time problems.

UNIT IV **15 Hours**

Web Mining, Mining the web page layout structure, mining web link structure, mining multimedia data on the web, Automatic classification of web documents and web usage mining; Distributed Data Mining.
Recent trends in Distributed Warehousing and Data Mining, Class Imbalance Problem; Graph Mining; Social Network Analysis.
Activities: Student presentation, Class discussion on different types of mining for the solution of real world problem.

Transactional Modes:

- Lecture
- E-tutorial
- Case Studies
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Han, J., & Kamber, M., (2011). Data Mining Concepts and Techniques. Elsevier Publication.
2. Tan, P., Kumar, V., and Steinbach M. (2016). Introduction to Data Minings. New Delhi: Pearson Education.
3. Dong, G., and Pei, J. (2007). Sequence Data Mining. New York: Springer.
4. Han, Jiawei, Kamber, Micheline, Pei, and Jian. (2012). Data mining: Concepts and techniques, USA: Morgan Kaufman publishers.
5. Kantardzic, M. (2011). Data mining: concepts, models, methods and algorithms. New Jersey: John, Wiley & sons.
6. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.553
Course Title: Intelligent Systems

Total Hours: 60

Course Objectives:
The aim of the course is to introduce to the field of Artificial Intelligence (AI) with emphasis on its use to solve real world problems for which solutions are difficult to express using the traditional algorithmic approach. It explores the essential theory behind methodologies for developing systems that demonstrate intelligent behaviour including dealing with uncertainty, learning from experience and following problem solving strategies found in nature.

Course Outcomes
After completion of course, students would be able to:

- Demonstrate knowledge of the fundamental principles of intelligent systems.
- Analyse and compare the relative merits of a variety of AI problem solving techniques.

UNIT I **15 Hours**
Search Methods Basic concepts of graph and tree search. Three simple search methods: breadth-first search, depth-first search, iterative deepening search. Heuristic search methods: best-first search, admissible evaluation functions, hill climbing search. Optimization and search such as stochastic annealing and genetic algorithm.

UNIT II **15 Hours**
Knowledge representation and logical inference Issues in knowledge representation. Structured representation, such as frames, and scripts, semantic networks and conceptual graphs. Formal logic and logical inference. Knowledge-based systems structures, its basic components. Ideas of Blackboard architectures.

UNIT III **15 Hours**
Reasoning under uncertainty and Learning Techniques on uncertainty reasoning such as Bayesian reasoning, Certainty factors and Dempster-Shafer Theory of Evidential reasoning, A study of different learning and evolutionary algorithms, such as statistical learning and induction learning.

UNIT IV **15 Hours**
Biological foundations to intelligent systems I: Artificial neural networks, Back propagation Networks, Radial basis function networks, and recurrent networks. Biological foundations to intelligent systems II: Fuzzy logic, knowledge Representation and inference mechanism, genetic algorithm, and fuzzy neural networks.
Recent trends in Fuzzy logic, Knowledge Representation

Transactional Modes:

- Lecture
- Peer Learning/Teaching
- E-tutorial
- Case Studies
- Online Teaching Tools

Suggested Readings:

1. Luger, G.F. and Stubblefield, W.A. (2001). Artificial Intelligence: Structures and strategies for Complex Problem Solving. United States: Addison Wesley.
2. Russell, S., and Norvig, P. (2015). Artificial Intelligence: A Modern Approach. New Delhi: Pearson Education India.
3. Russell S. and Norvig P. (2015). Artificial Intelligence: A Modern Approach. New Delhi: Pearson education India private limited.
4. Rich, E., Knight, K.N., Shivashankar, B. (2012). Artificial intelligence. New Delhi: Tata McGraw hill education private limited.
5. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.554
Course Title: Mobile Applications & Services

Total Hours: 60

Course Objectives:
This course presents the three main mobile platforms and their ecosystems, namely Android, iOS, and PhoneGap/Web OS. It explores emerging technologies and tools used to design and implement feature-rich mobile applications for smartphones and tablets

Course Outcomes
After completion of course, students would be able to:

- Explain the fundamentals, frameworks, and development lifecycle of mobile application platforms including iOS, Android, and PhoneGap.
- Identify the target platform and users.
- Design and develop a mobile application prototype in one of the platforms (challenge project).

UNIT I **15 Hours**

Introduction: Introduction to Mobile Computing, Introduction to Android Development Environment, Factors in Developing Mobile Applications, Mobile Software Engineering, Frameworks and Tools, Generic UI Development Android User.
Activities: Group Discussion, Case studies

UNIT II **15 Hours**

More on Uis: VUIs and Mobile Apps, Text-to-Speech Techniques, Designing the Right UI, Multichannel and Multimodal Uis, . Storing and Retrieving Data, Synchronization and Replication of Mobile Data, Getting the Model Right, Android Storing and Retrieving Data, Working with a Content Provider.
Activities: Assignment Based Learning, Live Demonstration

UNIT III **15 Hours**

Communications via Network and the Web: State Machine, Correct Communications Model, Android Networking and Web, Telephony Deciding Scope of an App, Wireless Connectivity and Mobile Apps, Android Telephony Notifications and Alarms-Performance, Performance and Memory Management, Android Notifications and Alarms, Graphics, Performance and Multithreading, Graphics and UI Performance, Android Graphics.
Activities: Implementation based Learning, Live Demonstrations of Android Notifications and Graphics

UNIT IV **15 Hours**

Putting It All Together: Packaging and Deploying, Performance Best Practices, Android Field Service App, Location Mobility and Location Based Services Android Multimedia: Mobile Agents and Peer-to-Peer Architecture, Android Multimedia Platforms and Additional Issues: Development Process, Architecture, Design, Technology Selection, Mobile App Development Hurdles, Testing,

Security and Hacking, Active Transactions, More on Security, Hacking Android. Recent trends in Communication protocols for IOT nodes, mobile computing techniques in IOT, agents based communications in IOT.
Activities: Case studies on recent trends, Presentations by students, Assignment based Learning.

Transactional Modes:

- Lecture
- Peer Learning/Teaching
- E-tutorial
- Experimentation
- Online Teaching Tools

Suggested Readings:

1. Lee, W. (2012). Beginning Android TM 4 Application Development. United Sates: John Wiley & Sons.
2. B'far, R.. (2013). Mobile computing principles: Designing and developing mobile applications with UML and XML. New Delhi: Cambridge university press.
3. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr

4	0	0	4

Course Code: CBS.552
Course Title: Cyber Threat Intelligence

Total Hours: 60

Course Objectives:
The outcome of this course is to introduce students to explain the cyber threats and cyber threat intelligence requirements. Classify cyber threat information and examine the potential for incidents and, provide more thoughtful responses.

Course Outcomes
After completion of course, students would be able to:
- Describe different Cyber Threat.
- Explain technique to Develop Cyber Threat Intelligence Requirements.
- Analyze and Disseminate Cyber Threat Intelligence

UNIT I **14 Hours**
Defining Cyber Threat Intelligence: The Need for Cyber Threat Intelligence: The menace of targeted attacks, The monitor-and-respond strategy, Why the strategy is failing, Cyber Threat Intelligence Defined, Key Characteristics: Adversary based, Risk focused, Process oriented, Tailored for diverse consumers, The Benefits of Cyber Threat Intelligence.
Activities: Case Study and Group Discussion

UNIT II **14 Hours**
Developing Cyber Threat Intelligence Requirements: Assets That Must Be Prioritized: Personal information, Intellectual property, Confidential business information, Credentials and IT systems information, Operational systems. Adversaries: Cybercriminals, Competitors and cyber espionage agents, Hacktivists. Intelligence Consumers: Tactical users, Operational users, Strategic users
Activities: Case study of real time social media cases

UNIT III **17 Hours**
Collecting Cyber Threat Information: Level 1: Threat Indicators, File hashes and reputation data, Technical sources: honeypots and scanners, Industry sources: malware and reputation feeds. Level 2: Threat Data Feeds, Cyber threat statistics, reports, and surveys, Malware analysis. Level 3: Strategic Cyber Threat Intelligence, Monitoring the underground, Motivation and intentions, Tactics, techniques, and procedures.
Analyzing and Disseminating Cyber Threat Intelligence: Information versus Intelligence, Validation and Prioritization: Risk scores, Tags for context, Human assessment. Interpretation and Analysis: Reports, Analyst skills, Intelligence platform, Customization. Dissemination: Automated feeds and APIs, Searchable knowledge base, Tailored reports.
Activities: Case study of real time social media cases

UNIT IV **15 Hours**
Selecting the Right Cyber Threat Intelligence Partner: Types of Partners: Providers of threat indicators, Providers of threat data feeds, Providers of comprehensive cyber threat intelligence. Important Selection Criteria: Global

and cultural reach, Historical data and knowledge, Range of intelligence deliverables, APIs and integrations, Intelligence platform, knowledge base, and portal, Client services, Access to experts. Intelligence-driven Security.
Activities: Flip Learning with Case studies of above concepts

Transactional Modes:

- Lecture cum Demonstration
- Cooperative learning
- Flipped classroom
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Friedman, J., and Bouchard, M., CISSP. Foreword by Watters, J. P., (1997). Definitive Guide to Cyber Threat Intelligence. Maryland: Cyber Edge Group, LLC.
2. Roberts, S. J., and Brown, R. (2017). Intelligence- Driven Incident Response: Outwitting the Adversary. California: O'Reilly Media.
3. Dalziel, H., (2014). How to Define and Build an Effective Cyber Threat Intelligence Capability. Elsevier Science & Technology.
4. Robertson, J., Diab, A., Marin, E., Nunes, E., Paliath, V., Shakarian, J., & Shakarian, P., (2017). DarkWeb Cyber Threat Intelligence Mining. New Delhi: Cambridge University Press.
5. Gourley, B., (2014). The Cyber Threat. United States: Createspace Independent Pub.
6. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.556
Course Title: Cost Management of Engineering Projects

Total Hours: 60

Course Objectives:
This course provides students with skills and knowledge of cost management of engineering projects. The course will enable students to understand the key components of engineering project.

Course Outcomes
After the completion of the course the students will be able to

- Employ their knowledge and skills together to understand the basics of a successful project.
- Explain the cost behaviour and profit planning
- Compare various quantitative methods for cost management

UNIT I **15 Hours**

Introduction and Overview of the Strategic Cost Management Process
Cost concepts in decision-making; Relevant cost, Differential cost, Incremental cost and Opportunity cost. Objectives of a Costing System; Inventory valuation; Creation of a Database for operational control; Provision of data for Decision-Making.
Activities: Numerical Example for above concepts

UNIT II **15 Hours**

Project: meaning, Different types, why to manage, cost overruns centers, various stages of project execution: conception to commissioning. Project execution as conglomeration of technical and nontechnical activities. Detailed Engineering activities. Pre project execution main clearances and documents Project team: Role of each member. Importance Project site: Data required with significance. Project contracts. Types and contents. Project execution Project cost control. Bar charts and Network diagram. Project commissioning: mechanical and process.
Activities: Case study of IT Companies

UNIT III **14 Hours**

Cost Behaviour and Profit Planning Marginal Costing; Distinction between Marginal Costing and Absorption Costing; Break-even Analysis, Cost-Volume-Profit Analysis. Various decision-making problems. Standard Costing and Variance Analysis. Pricing strategies: Pareto Analysis. Target costing, Life Cycle Costing. Costing of service sector. Just-in-time approach, Material Requirement Planning, Enterprise Resource Planning, Total Quality Management and Theory of constraints.
Activities: Case study and Numerical example to understand the above theory.

UNIT IV **15 Hours**

Activity-Based Cost Management, Bench Marking; Balanced Score Card and Value-Chain Analysis. Budgetary Control; Flexible Budgets; Performance

budgets; Zero-based budgets. Measurement of Divisional profitability pricing decisions including transfer pricing.
Quantitative techniques for cost management, Linear Programming, PERT/CPM, Transportation problems, Assignment problems, Simulation, Learning Curve Theory.
Activities: Case study and Numerical Example for better understanding.

Transactional Modes:

- Lecture
- E-tutorial
- Problem Solving
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Horngren, C. T., and Datar, S. M. (2017). Cost Accounting a Managerial Emphasis. New Delhi: Pearson Education.
2. Riahi-Belkaoui, A. (2001). Advanced Management Accounting. California: Greenwood Publication Group.
3. Kaplan, R. S., and Alkinson, A. A. (1998). Management Accounting. United States: Prentice Hall.
4. Bhattacharya, A. K. (2012). Principles & Practices of Cost Accounting. Allahabad, A. H. Wheeler publisher.
5. Vohra, N. D. (2017). Quantitative Techniques in Management. New Delhi: Tata McGraw Hill Education.
6. Rao, Thukaram M.E. (2011). Cost and management accounting. New Delhi: New age international publishers.
7. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CBS.553
Course Title: Cyber Law

Total Hours: 60

Course Objectives:
The objective of this course is to provide knowledge about the basic information on IT Act and Cyber law as well as the legislative and judicial development in the area.

Course Outcomes
After completion of course, students would be able to:

- Analyze fundamentals of Cyber Law.
- Discuss IT Act & its Amendments.
- Relate Cyber laws with security incidents.

UNIT I **Hours: 15**
Concept of Cyberspace, Issues of Jurisdiction in Cyberspace: Jurisdiction Principles under International law, Jurisdiction in different states, Position in India. Conflict of Laws in Cyberspace, International Efforts for harmonization Privacy in Cyberspace.
Activities: Case Studies on Jurisdiction

UNIT II **Hours: 15**
Electronic Commerce, Cyber Contract, Intellectual Property Rights and Cyber Laws. UNCITRAL Model Law, Digital Signature and Digital Signature Certificates, E-Governance and Records.
Activities: Brainstorming Sessions on Significance of UNCITRAL in day to day life of a common man.

UNIT III **Hours: 15**
Define Crime, Mens Rea, Crime in Context of Internet, Types of Cyber Crime, Computing Damage in Internet Crime, Offences under IPC (Indian Panel Code, 1860), Offences & Penalties under IT Act 2000, IT Act Amendments, Investigation & adjudication issues, Digital Evidence.
Activities: Exercises and problem solving skills on cyber disputes.

UNIT IV **Hours: 15**
Obscenity and Pornography, Internet and potential of Obscenity, International and National Instruments on Obscenity & Pornography, Child Pornography, Important Case Studies.
Activities: Exericses and problem solving skills on cyber crimes.

Transactional Modes:

- Lecture cum Demonstration
- Peer Learning/Teaching
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Ahmad, F. (2015). Cyber Law in India, Faridabad: New era law publications.
2. Sharma, J.P., Kanojia, S. (2016). Cyber Laws, New Delhi: Ane Books Pvt Ltd.
3. Chander, H. (2012). Cyber Laws and IT Protection. New Delhi: Prentice Hall India Learning Private Limited.
4. Justice Yatindra Singh. (2016). Cyber Laws. New Delhi: Universal Law Publishing Co.
5. Chaubey, R.K. (2012). An Introduction to cyber-crime and cyber law, Kolkata: Kamal Law House.
6. Tiwari, G. (2014). Understanding Laws: Cyber Laws & Cyber Crimes. New York: Lexis Nexis.
7. Seth, K. (2013). Justice Altamas Kabir, Computers Internet and New Technology Laws. New York: Lexis Nexis.
8. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
4	0	0	4

Course Code: CST.557
Course Title: Software Metrics

Total Hours: 60

Course Objectives:
Understand the underlying concepts, principles and practices in Software Measurements. Designing of Metrics model for software quality prediction and reliability.

Course Outcomes
After completion of course, students would be able to:
- Explain the role of software Metrics in Industry size software
- Prepare empirical investigation of software for a quality measurement
- Examine software reliability and problem solving by designing and selecting software reliability models.

UNIT I **15 Hours**
Overview of Software Metrics: Measurement in Software Engineering, Scope of Software Metrics, Measurement and Models Meaningfulness in measurement, Measurement quality, Measurement process, Scale, Measurement validation, Object-oriented measurements.
Goal based framework for software measurement: Software measure classification, Goal-Question-Metrics(GQM) and Goal-Question-Indicator-Metrics (GQIM), Applications of GQM and GQIM.
Activities: Case study and Group Discussion on OO methodology

UNIT II **15 Hours**
Empirical Investigation: Software engineering investigation, Investigation principles, Investigation techniques, Planning Formal experiments, Case Studies for Empirical investigations.
Object–oriented metrics: Object-Oriented measurement concepts, Basic metrics for OO systems, OO analysis and design metrics, Metrics for productivity measurement, Metrics for OO software quality.
Activities: Case study with Understand and Metrics Tools

UNIT III **15 Hours**
Measuring Internal Product attributes: Software Size, Length, reuse, Functionality, Complexity, Software structural measurement, Control flow structure, Cyclomatic Complexity, Data flow and data structure attributes Architectural measurement.
Measuring External Product attributes: Software Quality Measurements, Aspectes of Quality Measurements, Maintainability Measurements, Usability and Security Measurements.
Activities: Case study with Bugzila and JEERA tools

UNIT IV **15 Hours**

Measuring software Reliability: Concepts and definitions, Software reliability models and metrics, Fundamentals of software reliability engineering (SRE), Reliability management model.

Activities: Case study with Bugzila and JEERA tools

Transactional Modes:

- Lecture cum Demonstration
- Peer Learning/Teaching
- E-tutorial
- Self-Learning
- Online Teaching Tools

Suggested Readings:

1. Fenton, N. E. and Pfleeger, S. L. (1996). Software Metrics: A Rigorous and Practical Approach. New York: International Thomson Computer Press.
2. Kan, S. H. (2002). Metrics and Models in Software Quality Engineering. United States: Addison-Wesley Professional.
3. Anirban, B. (2015). Software Quality Assurance, Testing and Metrics. United States: Prentice Hall India Learning.
4. Tian, J. (2010). Software quality engineering: Testing, quality assurance and quantifiable improvement. New Delhi: Wiley India.
5. Research Articles from SCI & Scopus indexed Journals.

L	T	P	Cr
0	0	4	2

Course Code: CST.559
Course Title: Capstone Lab

In this, the student has to select an area and specify the base paper in that area to implement the same and show the results.
Evaluation criteria will be based on objectives stated and achieved

Course Objectives:
The objective of this lab is to help a team of students develop and execute an innovative project idea under the direction of the Capstone course Incharge.

Course Outcomes
After the completion of the course the students will be able to

- Apply the four phases of project development: requirements analysis, design, implementation, and documentation.

Timeline Work:

Month	AUG	SEP	NOV
Seminar	Submit area and Objectives to be achieved	Weekly report to faculty Incharge.	3rd week submit report 4th week Presentation

Evaluation Criteria:

Evaluation Parameter	Marks	Evaluated By
Area & Objectives	5	**Evaluation Committee**
Reports and Implementation	10	
Presentation and Viva-voce	10	
Total	**25**	

Student will be given final marks based the average marks by the Evaluation Committee

L	T	P	Cr
0	0	10	5

Course Code: CST.600
Course Title: Dissertation/Industrial Project

Course Objectives:

- The student shall have to write his/ her synopsis including an extensive review of literature with simultaneous identification of scientifically sound (and achievable) objectives backed by a comprehensive and detailed methodology. The students shall also present their synopsis to the synopsis approval committee.
- The second objective of Dissertation would be to ensure that the student learns the nuances of the scientific research. Herein the student shall have to carry out the activities/experiments to be completed during Dissertation (as mentioned in the synopsis).

Course Outcomes

The students would present their work to the Evaluation Committee (constituted as per the university rules). The evaluation criteria shall be as detailed below:

Course Contents
Evaluation criteria for Synopsis:

Evaluation Parameter	Marks	Evaluated by
Review of literature	50	**Internal Evaluation by Dean of School, HOD/ HOD nominee, Two faculty member nominated by Dean/HOD, Supervisor.**
Identification of gaps in knowledge and Problem Statement, Objective formulation & Methodology	50	
Total	**100**	

Student will be given final marks based the average marks by the Evaluation Committee

Timeline Works for Synopsis and Mid-Term:

Month	JULY	AUG	SEP	OCT	NOV	DEC
Synopsis	Bi- Weekly report submitted to Supervisor	Submission of Synopsis and Presentation				
Mid-Term			Bi- Weekly report submitted to Supervisor	Report submission in 3rd week Final Presentation in 4th week	Final Submission of Mid Term Report	

Grading of Marks:

Grades	A	B	C	D	E
Marks	85-100	84-75	74-65	64-40	0-40

Grading Evaluation:

Abbreviations of Grades	**Grades**
Excellent	A
Very Good	B
Good	C
Average	D
Below Average/ Un-Satisfactory	E

Evaluation criteria for Mid-Term:

Evaluation Parameter	**Maximum Marks**	**Evaluated By**
Mid Term Review and Presentation	50	**Internal/External Evaluation by Dean of School, HOD/ HOD nominee, Two faculty member nominated by Dean/ HOD, Supervisor.**
Continuous evaluation	50	
Total	**100**	

SEMESTER –IV

L	T	P	Cr
0	0	16	8

Course Code: CST.600
Course Title: Dissertation

Course Objectives:
In Dissertation the student shall have to carry out the activities/experiments to be completed during Dissertation (as mentioned in the synopsis).

Course Outcomes
The students would present their work to the evaluation Committee (constituted as per the university rules).
One research paper (either communicated to a Journal or accepted/ presented/published in conference proceedings) out of the dissertation research work is compulsory. The Evaluation criteria shall be as detailed below:

Evaluation Parameter	Maximum Marks	Evaluated By
Parameters by External Expert (As per University Criteria)	50	**Internal/External Evaluation by Dean of School, DAA Nominee, HOD/ HOD nominee, Supervisor.**
Presentation and defence of research work	50	
Total	100	

Student will be given final marks based the average marks by the Evaluation Committee

Timeline Work of Dissertation:

Month	**JAN**	**FEB**	**MAR**	**APR**	**MAY**	**JUN**
Dissertation	Bi- Weekly report submitted to Supervisor	Bi- Weekly report submitted to Supervisor	Report submission in 1st week	Pre- Submission Presentation in 3st week Report submission in 4th week	Final Submission of Dissertation/ Industrial Project and External Evaluation	

Grading of Marks:

Grades	A	B	C	D	E
Marks	85-100	84-75	74-65	64-40	0-40

Grading Evaluation:

Abbreviations of Grades	**Grades**
Excellent	A
Very Good	B
Good	C
Average	D
Below Average/ Un-Satisfactory	E