INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEEN DAYAL UPADHYAYA GORAKHPUR UNIVERSITY, GORAKHPUR

(दीन दयाल उपाध्याय गोरखपुर विश्वविद्यालय, गोरखपुर)

SYLLABUS

FOR

B. TECH.

INFORMATION TECHNOLOGY

BASED

ON

AICTE MODEL CURRICULUM

[Effective from the Session: 2023-24]

Mathematics for Machine Learning			
Course code	IT201		
Category	Professional Core Course		
Course title	Mathematics for Machine Learning (Theory)		
Scheme and Credits	**Credits**	3+0	Semester –III

Course Objectives:

- To understand the basic theory underlying machine learning.
- To be able to formulate machine learning problems corresponding to different applications.
- To understand a range of machine learning algorithms along with their strengths and weaknesses.
- To be able to apply machine learning algorithms to solve problems of moderate complexity.
- To apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models.

Unit-1	**Machine Learning:** Introduction to Machine Learning, Probability Theory, Model Selection, The Curse of Dimensionality, Decision Theory, Information Theory	**10 (Lectures)**
Unit-2	**Probability Distributions:** Binary Variables, Multinomial Variables, The Gaussian distribution, The Exponential Family, and Nonparametric Methods.	**8 (Lectures)**
Unit-3	**Linear Models for Regression:** Linear Basis Function Models, The Bias-Variance Decomposition, Bayesian Linear Regression, Bayesian Model Comparison, The Evidence Approximation, Limitations of Fixed Basis Functions	**10 (Lectures)**
Unit-4	Neural Networks and Kernel Methods	**8 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Explain theory underlying machine learning.
CO2 Construct algorithms to learn linear and non-linear models.
CO3 Implement data clustering algorithms.
CO4 Construct algorithms to learn tree and rule-based models.
CO5 Apply reinforcement learning techniques.

Textbooks & References:

1. Christopher M. Bishop. 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA.
2. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong,2020, Mathematics For Machine Learning. MIT-Press, Link: https://mml-book.com
3. Ethem Alpaydın -Introduction to Machine Learning Third Edition, MIT Press, 2004.
4. P. Flach, Machine Learning: The art and science of algorithms that make sense of data, Cambridge University Press, 2012.
5. K. P. Murphy, Machine Learning: A probabilistic perspective, MIT Press, 2012.
6. C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2007.
7. D. Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012.
8. M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning, MIT Press, 2012.
9. T. M. Mitchell, Machine Learning, McGraw Hill, 1997.

Data Structure				
Course code	IT202			
Category	Professional Core Course			
Course title	Data Structure (Theory)			
Scheme and Credits	**Credits**	4+0	Semester –III	
Course Objectives: An overview of data structure concepts, arrays, stack, queues, trees, and graphs. Discussion of various implementations of these data objects, programming styles, and run-time representations. Course also examines algorithms for sorting, searching and some graph algorithms.				
Unit-1	**Introduction:** Basic Terminology, Elementary Data Organization, Built in Data Types in C. Algorithm, Efficiency of an Algorithm, Time and Space Complexity, Asymptotic notations: Big Oh, Big Theta and Big Omega, Time-Space trade-off. Abstract Data Types (ADT) Arrays: Definition, Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order, Derivation of Index Formulae for 1-D,2-D,3-D and n-D Array Application of arrays, Sparse Matrices, and their representations. Linked lists: Array Implementation and Pointer Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal, Polynomial Representation and Addition Subtraction & Multiplications of Single variable & Two variables Polynomial.			**14 (Lectures)**
Unit-2	**Stacks and Queues:** Abstract Data Type, Primitive Stack operations: Push & Pop, Array and Linked Implementation of Stack in C, Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression, Iteration and Recursion- Principles of recursion, Tail recursion, Removal of recursion Problem solving using iteration and recursion with examples such as binary search, Fibonacci numbers, and Hanoi towers. Tradeoffs between iteration and recursion. Queues: Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Array and linked implementation of queues in C, Dequeue and Priority Queue.			**10 (Lectures)**
Unit-3	Searching & Sorting: Concept of Searching, Sequential search, Index Sequential Search, Binary Search. Concept of Hashing & Collision resolution Techniques used in Hashing. Sorting: Insertion Sort, Selection, Bubble Sort, Quick Sort, Merge Sort, Heap Sort and Radix Sort.			**10 (Lectures)**
Unit-4	**Trees:** Basic terminology used with Tree, Binary Trees, Binary Tree Representation: Array Representation and Pointer (Linked List) Representation, Binary Search Tree, Strictly Binary Tree, Complete Binary Tree. Extended Binary Trees, Tree Traversal algorithms: In-order, Preorder and Post-order, Constructing Binary Tree from given Tree Traversal, Operation of Insertion, Deletion, Searching & Modification of data in Binary Search. Threaded Binary trees, Traversing Threaded Binary trees. Huffman coding using Binary Tree. Concept & Basic Operations for AVL Tree, B Tree & Binary Heaps. **Graphs:** Terminology used with Graph, Data Structure for Graph Representations: Adjacency Matrices, Adjacency List, Adjacency. Graph Traversal: Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transitive Closure and Shortest Path algorithm: Warshal Algorithm and Dijikstra Algorithm.			**14 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Describe how arrays, linked lists, stacks, queues, trees, and graphs are represented in memory, used by the algorithms and their common applications.

CO2 Discuss the computational efficiency of the sorting and searching algorithms.

CO3 Implementation of Trees and Graphs and perform various operations on these data structure.

CO4 Understanding the concept of recursion, application of recursion and its implementation and removal of recursion.

CO5 Identify the alternative implementations of data structures with respect to its performance to solve a real-world problem.

Textbooks & References:

1. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data Structures Using C and C++", PHI Learning Private Limited, Delhi India
2. Horowitz and Sahani, "Fundamentals of Data Structures", Galgotia Publications Pvt Ltd Delhi India.
3. Lipschutz, "Data Structures" Schaum's Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd.
4. Thareja, "Data Structure Using C" Oxford Higher Education.
5. AK Sharma, "Data Structure Using C", Pearson Education India.
6. Rajesh K. Shukla, "Data Structure Using C and C++" Wiley Dreamtech Publication.
7. Michael T. Goodrich, Roberto Tamassia, David M. Mount "Data Structures and Algorithms in C++", Wiley India.
8. P. S. Deshpandey, "C and Data structure", Wiley Dreamtech Publication.
9. R. Kruse etal, "Data Structures and Program Design in C", Pearson Education.
10. Berztiss, AT: Data structures, Theory and Practice, Academic Press.
11. Jean Paul Trembley and Paul G. Sorenson, "An Introduction to Data Structures with applications", McGraw Hill.
12. Adam Drozdek "Data Structures and Algorithm in Java", Cengage Learning.

Python with Linux			
Course code	IT203		
Category	Professional Core Course		
Course title	Python with Linux (Theory)		
Scheme and Credits	**Credits**	2+0	Semester –III

Course Objectives:
An overview of data structure concepts, arrays, stack, queues, trees, and graphs. Discussion of various implementations of these data objects, programming styles, and run-time representations. Course also examines algorithms for sorting, searching and some graph algorithms.

Unit-1	**Introduction:** The Programming Cycle for Python, Python IDE, Interacting with Python Programs, Elements of Python, Type Conversion. Basics: Expressions, Assignment Statement, Arithmetic Operators, Operator Precedence, Boolean Expression. **Linux:** Introduction to Linux, Architecture of Linux, General Purpose Commands, Execution of codes on Linux.	**7 (Lectures)**
Unit-2	**Conditional Statements:** Conditional statement in Python (if-else statement, its working and execution), Nested-if statement and Elif statement in Python, Expression Evaluation & Float Representation. Loops: Purpose and working of loops, while loop including its working, For Loop, Nested Loops, Break and Continue.	**6 (Lectures)**
Unit-3	**Function:** Parts of A Function, Execution of A Function, Keyword and Default Arguments, Scope Rules. Strings: Length of the string and perform Concatenation and Repeat operations in it. Indexing and Slicing of Strings. Python Data Structure: Tuples, Unpacking Sequences, Lists, Mutable Sequences, List Comprehension, Sets, Dictionaries Higher Order Functions: Treat functions as first-class Objects, Lambda Expressions, Recursion.	**5 (Lectures)**
Unit-4	**Sieve of Eratosthenes:** generate prime numbers with the help of an algorithm given by the Greek Mathematician named Eratosthenes, whose algorithm is known as Sieve of Eratosthenes. File I/O: File input and output operations in Python Programming Exceptions and Assertions Modules: Introduction, Importing Modules, Abstract Data Types: Abstract data types and ADT interface in Python Programming. Classes: Class definition and other operations in the classes, Special Methods (such as _init_, _str_, comparison methods and Arithmetic methods etc.), Class Example, Inheritance, Inheritance and OOP. Searching and Sorting Techniques.	**6 (Lectures)**

Course Outcome (CO)
At the end of course, the student will be able to
CO1 Read and write simple Python programs.
CO2 Develop Python programs with conditionals and loops.
CO3 Define Python functions and to use Python data structures –- lists, tuples, dictionaries.
CO4 Do input/output with files in Python.
CO5 Do searching, sorting in Python.

Textbooks & References:

1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist ‘‘, 2nd edition, Updated for Python 3, Shroff/O ‘Reilly Publishers, 2016 (http://greenteapress.com/wp/thinkpython/)
2. Guido van Rossum and Fred L. Drake Jr, —An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.
3. 3.John V Guttag, —Introduction to Computation and Programming Using Python ‘‘, Revised and expanded Edition, MIT Press, 2013

4. 4.Robert Sedgewick, Kevin Wayne, Robert Dondero, —Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016.
5. 5.Timothy A. Budd, —Exploring Python||, Mc-Graw Hill Education (India) Private Ltd., 2015.
6. 6.Kenneth A. Lambert, —Fundamentals of Python: First Programs||, CENGAGE Learning, 2012.
7. 7.Charles Dierbach, —Introduction to Computer Science using Python: A Computational ProblemSolving Focus, Wiley India Edition, 2013.
8. 8.Paul Gries, Jennifer Campbell and Jason Montojo, —Practical Programming: An Introduction to Computer Science using Python 3, Second edition, Pragmatic Programmers, LLC, 2013.
9. Robert Love, "Linux Kernel Development", Pearson Education

Discrete Mathematics			
Course code	IT204		
Category	Professional Core Course		
Course title	Discrete Mathematics (Theory)		
Scheme and Credits	**Credits**	4+0	Semester –III

Course Objectives:
The objective of this course is to teach students how to think logically and mathematically. The course stresses on mathematical reasoning and describes different ways in which mathematical problems could be solved. There are four thematic areas covered in this course: mathematical reasoning, combinatorial analysis, discrete structures, and mathematical modeling.

Unit-1	**Set Theory:** Introduction, Combination of sets, Multisets, Ordered pairs. Proofs of some general identities on sets. **Relations:** Definition, Operations on relations, Properties of relations, Composite Relations, Equality of relations, Recursive definition of relation, Order of relations. **Functions:** Definition, Classification of functions, Operations on functions, recursively defined functions. Growth of Functions. **Natural Numbers:** Introduction, Mathematical Induction, Variants of Induction, Induction with Nonzero Base cases. Proof Methods, Proof by counter, Proof by contradiction.	**14 (Lectures)**
Unit-2	**Algebraic Structures:** Definition, Groups, Subgroups and order, Cyclic Groups, Cosets, Lagrange's theorem, Normal Subgroups, Permutation and Symmetric groups, Group Homomorphisms, Definition and elementary properties of Rings and Fields. **Lattices:** Definition, Properties of lattices – Bounded, Complemented, Modular and Complete lattice. **Boolean Algebra:** Introduction, Axioms and Theorems of Boolean algebra, Algebraic manipulation of Boolean expressions. Simplification of Boolean Functions, Karnaugh maps, Logic gates, Digital circuits and Boolean algebra.	**14 (Lectures)**
Unit-3	**Propositional Logic:** Proposition, well-formed formula, Truth tables, Tautology, Satisfiability, Contradiction, Algebra of proposition, Theory of Inference. **Predicate Logic:** First order predicate, well-formed formula of predicate, quantifiers, Inference theory of predicate logic.	**10 (Lectures)**
Unit-4	**Trees:** Definition, Binary tree, Binary tree traversal, Binary search tree. **Graphs:** Definition and terminology, Representation of graphs, Multigraphs, Bipartite graphs, Planar graphs, Isomorphism and Homeomorphism of graphs, Euler and Hamiltonian paths, Graph coloring, **Recurrence Relation & Generating function:** Recursive definition of functions, Recursive algorithms, Method of solving recurrences. **Combinatorics:** Introduction, Counting Techniques, Pigeonhole Principle	**10 (Lectures)**

Course Outcome (CO)
At the end of course, the student will be able to
CO1 Write an argument using logical notation and determine if the argument is or is not valid.
CO2 Understand the basic principles of sets and operations in sets.
CO3 Demonstrate an understanding of relations and functions and be able to determine their properties.
CO4 Demonstrate different traversal methods for trees and graphs.
CO5 Model problems in Computer Science using graphs and trees.

Textbooks & References:

1. Koshy, Discrete Structures, Elsevier Pub. 2008 Kenneth H. Rosen, Discrete Mathematics and Its Applications, 6/e, cGraw-Hill, 2006.
2. B. Kolman, R.C. Busby, and S.C. Ross, Discrete Mathematical Structures, 5/e, Prentice Hall, 2004.
3. E. Scheinerman, Mathematics: A Discrete Introduction, Brooks/Cole, 2000.
4. R.P. Grimaldi, Discrete and Combinatorial Mathematics, 5/e, Addison Wesley, 2004
5. Liptschutz, Seymour, "Discrete Mathematics", McGraw Hill.
6. Trembley, J.P & R. Manohar, "Discrete Mathematical Structure with Application to Computer Science", McGraw Hill.
7. Deo, Narsingh, "Graph Theory With application to Engineering and Computer Science.", PHI.
8. Krishnamurthy, V., "Combinatorics Theory & Application", East-West Press Pvt. Ltd., New Delhi

Mathematics for Machine Learning Lab			
Course code	IT251		
Category	Professional Core Course		
Course title	Mathematics for Machine Learning Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –III

EXP-1 Implementation of Manhatton and Euclidean Distance in Python.
EXP-2 Implementation of Jaccord Index in Python for given dataset.
EXP-3 Implementation of Cosine Similarity in Python for given dataset.
EXP-4 Apply linear regression on given dataset.
EXP-5 Apply multiple regression on given dataset.
EXP-6 Implement "AND" gate using McCulloch-Pitts network.
EXP-7 Implement "OR" gate using McCulloch-Pitts network.
EXP-8 Implement "NOT" gate using McCulloch-Pitts network.
EXP-9 Implement "NOR" gate using McCulloch-Pitts network.
EXP-10 Implement "NAND" gate using McCulloch-Pitts network.
EXP-11 Implement "XOR" gate using McCulloch-Pitts network.
EXP-12 Implement "AND" gate using Hebb training algorithm.

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Data Structure Lab			
Course code	IT252		
Category	Professional Core Course		
Course title	Data Structure Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –III

Write C Programs to illustrate the concept of the following:

EXP-1 Searching Algorithm.
EXP-2 Sorting Algorithms-Non-Recursive and Recursive.
EXP-3 Implementation of Stack using Array.
EXP-4 Implementation of Queue using Array.
EXP-5 Implementation of Circular Queue using Array.
EXP-6 Implementation of Stack using Linked List.
EXP-7 Implementation of Queue using Linked List.
EXP-8 Implementation of Circular Queue using Linked List.
EXP-9 Implementation of Tree Structures, Binary Tree, Tree Traversal, Binary Search Tree, Insertion and Deletion in BST.
EXP-10 Graph Implementation, BFS, DFS, Minimum cost spanning tree, shortest path algorithm.

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Python with Linux Lab			
Course code	IT253		
Category	Professional Core Course		
Course title	Python with Linux Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –III

EXP-1 Krishna and his five friends have decided to go for an industrial visit by sharing the expenses of the fuel equally. Write a Python program to calculate the amount (in Rs) each of them needs to put in for the complete journey. The program should also display True, if the amount to be paid by each person is divisible by 3, otherwise it should display False.
Hint: Use the relational operators in print statement.
Assumptions: Assume that mileage of the vehicle, amount per liter of fuel and distance for one way are given.

EXP-2 Write a python program to find the best of two test average marks out of three test's marks accepted from the user.

EXP-3 Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number.

EXP-4 Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed.

EXP-5 Develop a python program to convert binary to decimal, octal to hexadecimal using functions.

EXP-6 Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters.

EXP-7 Write a python program to implement insertion sort and merge sort using lists.

EXP-8 Write a program to convert roman numbers in to integer values using dictionaries.

EXP-9 Write a function called isphonenumber () to recognize a pattern 415-555-4242 without using regular expression and also write the code to recognize the same pattern using regular expression.

EXP-10 Develop a python program that could search the text in a file for phone numbers (+919900889977) and email addresses (sample@gmail.com).

12. Write a python program to accept a file name from the user and perform the following operations:
    a) Display the first N line of the file.
    b) Find the frequency of occurrence of the word accepted from the user in the file.

EXP-11 Write a python program to create a ZIP file of a particular folder which contains several files inside it.

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Introduction to Evolutionary Computing			
Course code	IT205		
Category	Professional Core Course		
Course title	Introduction to Evolutionary Computing (Theory)		
Scheme and Credits	**Credits**	3+0	Semester –IV

Course Objectives:

To provide a broad introduction to the field of Genetic Algorithms and other fields of Evolutionary Computation and global optimization. To teach students how to apply these methods to solve problems in complex domains. The course is appropriate both for students preparing for research in Evolutionary Computation, as well as Science and Engineering students who want to apply Evolutionary Computation techniques to solve problems in their fields of study.

Unit-1	**Introduction to Evolutionary Computation:** Biological and artificial evolution, evolutionary computation and AI, different historical branches of EC, a simple genetic algorithm. **Search Operators:** Crossover, mutation, crossover and mutation rates, Crossover for real-valued representations, mutation for real-valued representations, combinatorial GA.	**9 (Lectures)**
Unit-2	**Selection Schemes:** Fitness proportional selection and fitness scaling, ranking, tournament selection, selection pressure and its impact on evolutionary search. **Theoretical Analysis of Evolutionary Algorithms:** Schema theorems, convergence of the algorithms, computational time complexity of the algorithms, no free lunch theorem.	**9 (Lectures)**
Unit-3	**Search Operators and Representations:** Mixing different search operators, adaptive representations. Niching and Speciation: Fitness sharing, crowding and mating restriction. **Constraint Handling:** Common techniques, penalty methods, repair methods, Deb's penalty parameter method.	**9 (Lectures)**
Unit-4	**Multi-objective Evolutionary Optimization:** Pareto optimality, multi-objective evolutionary algorithms: MOGA, NSGA-II, etc. Applications of GA in engineering problems, job-shop scheduling and routing problems.	**9 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Complete a significant programming project that uses an evolutionary programming technique.

CO2 Solve a problem that would be difficult, if not impossible to solve without a self-adaptive approach.

CO3 Characterize genetic programming, genetic algorithms, and evolutionary programming, and specify the conditions under which these techniques might be most applicable.

CO4 Compare and contrast evolutionary strategies and genetic programming against canonical genetic algorithms in terms of representation, selection, crossover, and mutation.

CO5 Write a brief essay explaining why the Scheme Theorem may explain why evolutionary techniques can succeed in some situations, and why they may fail in others.

Textbooks & References:

1. Goldberg D.E. Genetic Algorithms in Search, Optimization and Machine Learning. Pearson Education Asia 2002
2. K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley and Sons, 2009.
3. M. Mitchell, An introduction to genetic algorithms, MIT Press, 1996.
4. L. D. Davis, Evolutionary algorithms, Springer-Verlag, 1999.
5. K. Srinivasa Raju and D. Nagesh Kumar. Multicriterion Analysis in Engineering and Management. PHI Learning Pvt. Ltd., New Delhi, India 2010.

Computer Architecture & Organization				
Course code	IT206			
Category	Professional Core Course			
Course title	Computer Architecture & Organization(Theory)			
Scheme and Credits	**Credits**	3+0	Semester –IV	

Course Objectives:
The objectives for this course are:
- To understand the structure, function and characteristics of computer systems.
- To understand the design of the various functional units and components of computers.
- To identify the elements of modern instructions sets and their impact on processor design.

Unit-1	**Introduction:** Functional units of digital system and their interconnections, bus architecture, types of buses and bus arbitration. Register, bus and memory transfer. **Processor organization:** general registers organization, stack organization and addressing modes. Arithmetic and logic unit, Look ahead carries adders. **Multiplication:** Signed operand multiplication, Booth's algorithm and array multiplier, Division and logic operations, Floating point arithmetic operation, Arithmetic & logic unit design, IEEE Standard for Floating Point Numbers.	**9 (Lectures)**
Unit-2	**Control Unit:** Instruction types, formats, instruction cycles and sub cycles (fetch and execute etc.), micro-operations, execution of a complete instruction, Program Control, Reduced Instruction Set Computer, Pipelining, Hardwire and micro programmed control, micro programme sequencing, concept of horizontal and vertical microprogramming	**9 (Lectures)**
Unit-3	**Memory:** Basic concept and hierarchy, semiconductor RAM memories, 2D & 2 1/2D memory organization. ROM memories. Cache memories: concept and design issues & performance, address mapping and replacement **Auxiliary Memories:** magnetic disk, magnetic tape and optical disks, Virtual memory concept implementation.	**9 (Lectures)**
Unit-4	**Input / Output Organization:** Peripheral devices, I/O interface, I/O ports, **Interrupts:** interrupt hardware, types of interrupts and exceptions. **Modes of Data Transfer:** Programmed I/O, interrupt initiated I/O and Direct Memory Access, I/O channels, and processors. **Serial Communication:** Synchronous & asynchronous communication, standard communication interfaces.	**9 (Lectures)**

Course Outcomes (CO)
At the end of course, student will be able to
CO1 Study of the basic structure and operation of a digital computer system.
CO2 Analysis of the design of arithmetic & logic unit and understanding of the fixed point and floating-point arithmetic operations.
CO3 Implementation of control unit techniques and the concept of Pipelining.
CO4 Understanding the hierarchical memory system, cache memories and virtual memory.
CO5 Understanding the different ways of communicating with I/O devices and standard I/O interfaces.

Textbooks & References:
1. Computer System Architecture - M. Mano.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky Computer Organization, McGraw-Hill, Fifth Edition, Reprint 2012.
3. John P. Hayes, Computer Architecture and Organization, Tata McGraw Hill, Third Edition, 1998. Reference books.
4. William Stallings, Computer Organization and Architecture-Designing for Performance, Pearson

Education, Seventh edition, 2006.
5. Behrooz Parahami, "Computer Architecture", Oxford University Press, Eighth Impression, 2011.
6. David A. Patterson and John L. Hennessy, "Computer Architecture-A Quantitative Approach", Elsevier, a division of reed India Private Limited, Fifth edition, 2012.
7. Structured Computer Organization, Tannenbaum (PHI.)

Introduction to Operating System				
Course code	IT207			
Category	Professional Core Course			
Course title	Introduction to Operating System (Theory)			
Scheme and Credits	**Credits**	4+0	Semester –IV	

Course Objectives:

- To understand the services provided by and the design of an operating system.
- To understand the structure and organization of the file system.
- To understand what a process is and how processes are synchronized and scheduled.
- To understand different approaches to memory management.

Unit-1	**Operating Systems Overview:** Components, Goals of Designer, System Structures, User Services, Interrupt Systems and Device Programming- Interrupt Sources and Priorities, Interrupt Service Routines, Hardware Support - Machine States, Context Switching, Privileged Instructions and Registers.	**12 (Lectures)**
Unit-2	**Memory Management-Major Issues:** Fetch, Placement, Contiguity, Relocation Adjustment, Paging and Virtual Memory, Translate-Look-Aside Buffer (Associative Memory), Single and Multi-Level Page Tables, Paging with Segmentation, Problems of Large Address Spaces and How They Are Addressed Virtual Storage Management- Storage Hierarchy, Cache Usage, Partial Residency, Page Replacement Strategies, Working Sets.	**12 (Lectures)**
Unit-3	**Concurrency Problems and Solutions:** Critical Section Problem, Process Synchronization and Coordination, Semaphores, Special Instructions, Monitors, Inter-process Communication, Remote Procedure Calls, Special Problems of Transaction-Based Systems Deadlock and Resource Conflict-Prevention, Avoidance, Detection, Recovery, Process and Thread Management-Process/Thread Creation and Termination, Process/Thread States and Their Transitions CPU Scheduling Algorithms, Non-Preemptive Approaches, Preemptive Approach, Multi-Processor Considerations.	**12 (Lectures)**
Unit-4	**Physical Storage Management:** Disk Scheduling Algorithms, Disk Performance Features, Disk Reliability Concerns File System Organization, The Boot Record, Where Things Start, Directory Organization, File Descriptors, Access Control Backup System Security-Principle of Least Privilege, Threats and Vulnerabilities. **Protection Mechanisms:** Access and Capability Control, User (Subject) Authentication, Levels of Security in "Trusted" Systems, Confinement Problem.	**12 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Understand the structure and functions of OS.
CO2 Learn about Processes, Threads and Scheduling algorithms.
CO3 Understand the principles of concurrency and Deadlocks.
CO4 Learn various memory management scheme.
CO5 Study I/O management and File systems.

Textbooks & References:

1. Silberschatz, Galvin and Gagne, “Operating Systems Concepts”, Wiley.
2. Sibsankar Halder and Alex A Aravind, “Operating Systems”, Pearson Education.
3. Harvey M Dietel, “An Introduction to Operating System”, Pearson Education.
4. D M Dhamdhere, “Operating Systems: A Concept based Approach”, 2nd Edition, TMH.
5. William Stallings, “Operating Systems: Internals and Design Principles”, 6th Edition, Pearson Education

Software Engineering			
Course code	IT208		
Category	Professional Core Course		
Course title	Software Engineering (Theory)		
Scheme and Credits	**Credits**	3+0	Semester –IV

Course Objectives:

- To provide the idea of decomposing the given problem into Analysis, Desing, Implementation, Testing and Maintenance phases.
- To provide an idea of using various process models in the software industry according to given circumstances.
- To gain the knowledge of how Analysis, Design, Implementation, Testing and Maintenance processes are conducted in a software project.

Unit-1	**Software Process:** Introduction, Software Engineering Paradigm, Life Cycle Models (Waterfall, Incremental, Spiral, Evolutionary, Prototyping), Software Requirements, Functional and Non-Functional, Software Document, Requirement Engineering Process, Feasibility Studies, **Software Prototyping:** Prototyping in Software, Process Data Functional and Behavioral Models, Structured Analysis and Data Dictionary.	**9 (Lectures)**
Unit-2	**Basic Concept of Software Design, Architectural Design, Low Level Design:** Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design. **Software Measurement and Metrics:** Various Size Oriented Measures, Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures, Control Flow Graph.	**9 (Lectures)**
Unit-3	**Software Testing:** Taxonomy of S/W Testing Levels, Black Box Testing, Testing Boundary Conditions, Structural Testing, Regression Testing, S/W Testing Strategies, Unit Testing, Integration Testing, Validation Testing, System Testing and Debugging.	**9 (Lectures)**
Unit-4	**Measures and Measurements**: Zipf's Law, Software Cost Estimation, Function Point Models, COCOMO Model. Delphi Method Scheduling, Earned Value Analysis, Error Tracking, Software Configuration Management, Program Evolution Dynamics, Software Maintenance, Project Planning, Project Scheduling, Risk Management, Case Tools.	**9 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Explain various software characteristics and analyze different software Development Models.

CO2 Demonstrate the contents of a SRS and apply basic software quality assurance practices to ensure that design, development meet or exceed applicable standards.

CO3 Compare and contrast various methods for software design.

CO4 Formulate testing strategy for software systems, employ techniques such as unit testing, Test driven development and functional testing.

CO5 Manage software development process independently as well as in teams and make use of Various software management tools for development, maintenance and analysis.

Textbooks & References:

1. RS Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. Pankaj Jalote, Software Engineering, Wiley.
3. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.
4. KK Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.
5. Ghezzi, M. Jarayeri, D. Manodrioli, Fundamentals of Software Engineering, PHI Publication.

6. Ian Sommerville, Software Engineering, Addison Wesley.
7. Kassem Saleh, "Software Engineering", Cengage Learning.
8. P fleeger, Software Engineering, Macmillan Publication.

Automata and Formal Languages				
Course code	IT209			
Category	Professional Core Course			
Course title	Automata and Formal Languages (Theory)			
Scheme and Credits	**Credits**	4+0	Semester –IV	

Course Objectives:

The objective of this course is to explore the theoretical foundations of computer science from the perspective of formal languages and classify machines by their power to recognize languages.

Unit-1	**Introduction:** Alphabets, Strings and Languages; Automata and Grammars, Deterministic finite Automata (DFA)-Formal Definition, **Simplified Notation:** State transition graph, Transition table, Language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition, Language of NFA, Equivalence of NFA and DFA, Minimization of Finite Automata, Distinguishing one string from other, Myhill-Nerode Theorem.	**9 (Lectures)**
Unit-2	**Regular expression (RE):** Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleen's Theorem, Regular expression to FA, DFA to Regular expression, Arden Theorem, Non-Regular Languages, Pumping Lemma for regular Languages, Application of Pumping Lemma, Closure properties of Regular Languages, Decision properties of Regular Languages, FA with output: Moore and Mealy machine, Equivalence of Moore and Mealy Machine, Applications and Limitation of FA.	**9 (Lectures)**
Unit-3	**Context Free Grammar (CFG) and Context Free Languages (CFL):** Definition, Examples, Derivation, Derivation trees, Ambiguity in Grammar, Inherent ambiguity, Ambiguous to Unambiguous CFG, Useless symbols, Simplification of CFGs, Normal forms for CFGs: CNF and GNF, Closure properties of CFLs, Decision Properties of CFLs: Emptiness, Finiteness and Membership, Pumping lemma for CFLs, **Push Down Automata (PDA):** Description and definition, Instantaneous Description, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic PDA, Equivalence of PDA and CFG, CFG to PDA and PDA to CFG.	**9 (Lectures)**
Unit-4	**Basic Model:** definition and representation, Instantaneous Description, Language acceptance by TM, Variants of Turing Machine, TM as Computer of Integer functions, Universal TM, Church's Thesis, Recursive and recursively enumerable languages, Halting problem, Introduction to Undecidability, Undecidable problems about TMs. Post correspondence problem (PCP), Modified PCP, Introduction to recursive function theory.	**9 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Analyse and design finite automata, pushdown automata, Turing machines, formal languages, and grammars.

CO2 Analyse and design, Turing machines, formal languages, and grammars.

CO3 Demonstrate the understanding of key notions, such as algorithm, computability, decidability, and complexity through problem solving.

CO4 Prove the basic results of the Theory of Computation.

CO5 State and explain the relevance of the Church-Turing thesis.

Textbooks & References:

1. Introduction to Automata theory, Languages and Computation, J.E.Hopcraft, R.Motwani, and Ullman. 2nd edition, Pearson Education Asia.
2. Introduction to languages and the theory of computation, J Martin, 3rd Edition, Tata McGraw Hill.

3. Elements and Theory of Computation, C Papadimitrou and C. L. Lewis, PHI .
4. Mathematical Foundation of Computer Science, Y.N.Singh, New Age International.
5. An Introduction to Formal Language and Automata, Peter Linz, Narosa Pub. House.

Evolutionary Computing Lab			
Course code	IT255		
Category	Professional Core Course		
Course title	Evolutionary Computing Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –IV

EXP-1 Neural Network Implementation

- **a)** Implement Perceptron Network
- **b)** Implement Adaline Network
- **c)** Implement Madaline Network for XOR Function
- **d)** Implement Back Propagation Network for XOR Function using Bipolar Inputs and Binary Targets.
- **e)** Implement Kohonen Self-Organizing Feature Map

EXP-2 Genetic Algorithm Implementation

- **a)** To Maximize F(X1, X2)=4X1+3X2
- **b)** To minimize F(X)=X2
- **c)** Implementation of Traveling Salesman Problem
- **d)** To find the roots of Quadratic Equation

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Computer Architecture & Organization Lab			
Course code	IT256		
Category	Professional Core Course		
Course title	Computer Architecture & Organization Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –IV

EXP-1 Implementing HALF ADDER, FULL ADDER using basic logic gates
EXP-2 Implementing Binary -to -Gray, Gray -to -Binary code conversions.
EXP-3 Implementing 3-8 line DECODER.
EXP-4 Implementing 4x1 and 8x1 MULTIPLEXERS.
EXP-5 Verify the excitation tables of various FLIP-FLOPS.
EXP-6 Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
EXP-7 Design of an 8-bit ARITHMETIC LOGIC UNIT.
EXP-8 Design the data path of a computer from its register transfer language description.
EXP-9 Design the control unit of a computer using either hardwiring or microprogramming based on its register transfer language description.
EXP-10 Implement a simple instruction set computer with a control unit and a data path.

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Operating System Lab			
Course code	IT257		
Category	Professional Core Course		
Course title	Operating System Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –IV

EXP-1 Study of hardware and software requirements of different operating systems (UNIX, LINUX, WINDOWS XP, WINDOWS7/8/10/11.

EXP-2 Execute various UNIX system calls for a. Process management b. File management c. Input/output Systems calls.

EXP-3 Implement CPU Scheduling Policies:

c) SJF
d) Priority
e) FCFS
f) Multi-level Queue.

EXP-4 Implement file storage allocation technique:

a) Contiguous (using array)
b) Linked –list (using linked-list)
c) Indirect allocation (indexing).

EXP-5 Implementation of contiguous allocation techniques:

a) Worst-Fit
b) Best- Fit
c) First- Fit

EXP-6 Calculation of external and internal fragmentation a. Free space list of blocks from system b. List process file from the system.

EXP-7 Implementation of compaction for the continually changing memory layout and calculate total movement of data.

EXP-8 Implementation of resource allocation graph RAG).

EXP-9 Implementation of Banker‟s algorithm.

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Software Engineering Lab			
Course code	IT258		
Category	Professional Core Course		
Course title	Software Engineering Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –IV

Do the following 7 exercises for any two projects given in the list of sample projects or any other projects:

EXP-1 Development of problem statement.
EXP-2 Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
EXP-3 Preparation of Software Configuration Management and Risk Management related documents.
EXP-4 Study and usage of any Design phase CASE tool
EXP-5 Performing the Design by using any Design phase CASE tools.
EXP-6 Develop test cases for unit testing and integration testing
EXP-7 Develop test cases for various white box and black box testing techniques.

Sample Projects (Other Projects can be also assigned):

a) Passport automation System
b) Book Bank
c) Online Exam Registration
d) Stock Maintenance System
e) Online course reservation system
f) E-ticketing
g) Software Personnel Management System
h) Credit Card Processing
i) E-book management System.
j) Recruitment system

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Artificial Intelligence				
Course code	IT301			
Category	Professional Core Course			
Course title	Artificial Intelligence (Theory)			
Scheme and Credits	**Credits**	4+0	Semester –V	
Course Objectives: This course sheds light on the fundamental of Artificial Intelligence and its applications in various areas.				
Unit-1	**Introduction:** Definition, Future of Artificial Intelligence, Characteristics of Intelligent Agents, Typical Intelligent Agents, Problem Solving Approach to Typical AI problems.			**12 (Lectures)**
Unit-2	**Problem Solving Methods:** Problem solving Methods, Search Strategies, Uninformed, Informed, Heuristics, Local Search Algorithms and Optimization Problems, Searching with Partial Observations, Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search, Game Playing, Optimal Decisions in Games, Alpha, Beta Pruning, Stochastic Games.			**12 (Lectures)**
Unit-3	**Knowledge Representation:** First Order Predicate Logic, Prolog Programming, Unification, Forward Chaining, Backward Chaining, Resolution, Knowledge Representation, Ontological Engineering, Categories and Objects, Events, Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.			**12 (Lectures)**
Unit-4	**Software Agents & Applications**: Architecture for Intelligent Agents, Agent communication, Negotiation and Bargaining, Argumentation among Agents, Trust and Reputation in Multi-agent systems. **APPLICATIONS:** AI applications, Language Models, Information Retrieval, Information Extraction, Natural Language Processing, Machine Translation, Speech Recognition, Robot, Hardware, Perception, Planning, Moving.			**12 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Understand the basics of the theory and practice of Artificial Intelligence as a discipline and about intelligent agents.

CO2 Understand search techniques and gaming theory.

CO3 Apply knowledge representation techniques and problem-solving strategies to common AI applications.

CO4 Aware of techniques used for classification and clustering.

CO5 Aware of basics of pattern recognition and steps required for it.

Textbooks & References:

1. S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach||, Prentice Hall, Third Edition, 2009.
2. I. Bratko, "Prolog: Programming for Artificial Intelligence", Fourth edition, Addison-Wesley Educational Publishers Inc., 2011.
3. M. Tim Jones, —Artificial Intelligence: A Systems Approach, Jones and Bartlett Publishers, Inc.First Edition, 2008
4. Nils J. Nilsson, —The Quest for Artificial Intelligence, Cambridge University Press, 2009.
5. William F. Clocksin and Christopher S. Mellish, Programming in Prolog: Using the ISO Standard, Fifth Edition, Springer, 2003.
6. Gerhard Weiss, —Multi Agent Systems, Second Edition, MIT Press, 2013.
7. David L. Poole and Alan K. Mackworth, —Artificial Intelligence: Foundations of Computational Agents, Cambridge University Press, 2010.

Database Management System				
Course code	IT302			
Category	Professional Core Course			
Course title	Database Management System (Theory)			
Scheme and Credits	**Credits**	3+0	Semester –V	

Course Objectives:

- To explain basic database concepts, applications, data models, schemas, and instances.
- To demonstrate the use of constraints and relational algebra operations.
- To emphasize the importance of normalization in databases.
- To facilitate students in Database design
- To familiarize issues of concurrency control and transaction management

Unit-1	**Introduction:** Overview, Database System vs File System, Database System Concept and Architecture, Data Model Schema and Instances, Data Independence and Database Language and Interfaces, Data Definitions Language, DML, Overall Database Structure. Data Modeling Using the Entity Relationship Model: ER Model Concepts, Notation for ER Diagram, Mapping Constraints, Keys, Concepts of Super Key, Candidate Key, Primary Key, Generalization, Aggregation, Reduction of an ER Diagrams to Tables, Extended ER Model, Relationship of Higher Degree.	**9 (Lectures)**
Unit-2	**Relational Data Model and Language:** Relational Data Model Concepts, Integrity Constraints, Entity Integrity, Referential Integrity, Keys Constraints, Domain Constraints, Relational Algebra, Relational Calculus, Tuple and Domain Calculus. **Introduction on SQL:** Characteristics of SQL, Advantage of SQL, SQL Data Type and Literals, Types of SQL Commands, SQL Operators and Their Procedure, Tables, Views and Indexes, Queries and Sub Queries, Aggregate Functions, Insert, Update and Delete Operations, Joins, Unions, Intersection, Minus, Cursors, Triggers, Procedures in SQL/PL SQL.	**9 (Lectures)**
Unit-3	**Data Base Design & Normalization:** Functional dependencies, Normal forms, First normal form, Second normal form, Third normal form, BCNF, Inclusion dependence, Loss less join decompositions, Normalization using FD, MVD, and JDs, Alternative approaches to database design.	**8 (Lectures)**
Unit-4	**Transaction Processing Concept:** Transaction System, Testing of Serializability, Serializability of Schedules, Conflict & View Serializable Schedule, Recoverability, Recovery from Transaction Failures, Log Based Recovery, Checkpoints, Deadlock Handling. **Distributed Database:** Distributed Data Storage, Concurrency Control, Directory System. **Concurrency Control Techniques:** Concurrency Control, Locking Techniques for Concurrency Control, Time Stamping Protocols for Concurrency Control, Validation Based Protocol, Multiple Granularity, Multi Version Schemes, Recovery with Concurrent Transaction, Case Study of Oracle.	**10 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Apply knowledge of database for real life applications.

CO2 Apply query processing techniques to automate the real time problems of databases.

CO3 Identify and solve the redundancy problem in database tables using normalization.

CO4 Understand the concepts of transactions, their processing so they will be familiar with a broad range of database management issues including data integrity, security and recovery.

CO5 Design, develop and implement a small database project using database tools.

Textbooks & References:

1. Korth, Silbertz, Sudarshan," Database Concepts", McGraw Hill
2. Date C J, "An Introduction to Database Systems", Addision Wesley
3. Elmasri, Navathe, "Fundamentals of Database Systems", Addision Wesley
4. O'Neil, Databases, Elsevier Pub.
5. Ramakrishnan"Database Management Systems",McGraw Hill

Design and Analysis of Algorithm			
Course code	IT303		
Category	Professional Core Course		
Course title	Design and Analysis of Algorithm (Theory)		
Scheme and Credits	**Credits**	4+0	Semester –V

Course Objectives:
The goal of this course is to provide a solid background in the design and analysis of the major classes of algorithms. At the end of the course students will be able to develop their own versions for a given computational task and to compare and contrast their performance.

Unit-1	**Introduction:** Algorithms, Analyzing Algorithms, Complexity of Algorithms, Growth of Functions, Performance Measurements, Sorting and Order Statistics - Shell Sort, Quick Sort, Merge Sort, Heap Sort, Comparison of Sorting Algorithms, Sorting in Linear Time.	**12 (Lectures)**
Unit-2	**Advanced Data Structures:** Red-Black Trees, B – Trees, Binomial Heaps, Fibonacci Heaps, Tries, Skip List.	**12 (Lectures)**
Unit-3	**Divide and Conquer with Examples** Such as Sorting, Matrix Multiplication, Convex Hull and Searching. **Greedy Methods with Examples** Such as Optimal Reliability Allocation, Knapsack, Minimum Spanning Trees – Prim's and Kruskal's Algorithms, Single Source Shortest Paths - Dijkstra's and Bellman Ford Algorithms.	**12 (Lectures)**
Unit-4	**Dynamic Programming with Examples** Such as Knapsack, All Pair Shortest Paths – Warshal's and Floyd's Algorithms, Resource Allocation Problem. Backtracking, **Branch and Bound with Examples** Such as Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cycles and Sum of Subsets.	**12 (Lectures)**

Course Outcome (CO)
At the end of course, the student will be able to
CO1 Design new algorithms, prove them correct, and analyze their asymptotic and absolute runtime and memory demands.
CO2 Find an algorithm to solve the problem (create) and prove that the algorithm solves the problem correctly (validate).
CO3 Understand the mathematical criterion for deciding whether an algorithm is efficient and knows many practically important problems that do not admit any efficient algorithms.
CO4 Apply classical sorting, searching, optimization and graph algorithms.
CO5 Understand basic techniques for designing algorithms, including the techniques of recursion, divide-and-conquer, and greedy.

Textbooks & References:

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, "Introduction to Algorithms", Printice Hall of India.
2. E. Horowitz & S Sahni, "Fundamentals of Computer Algorithms",
3. Aho, Hopcraft, Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education, 2008.
4. LEE "Design & Analysis of Algorithms (POD)", McGraw Hill
5. Richard E.Neapolitan "Foundations of Algorithms" Jones & Bartlett Learning
6. Jon Kleinberg and Éva Tardos, Algorithm Design, Pearson, 2005.
7. Michael T Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples,Second Edition, Wiley, 2006.
8. Harry R. Lewis and Larry Denenberg, Data Structures and Their Algorithms, Harper Collins, 1997
9. Robert Sedgewick and Kevin Wayne, Algorithms, fourth edition, Addison Wesley, 2011.

10. Harsh Bhasin,"Algorithm Design and Analysis",First Edition,Oxford University Press.
11. Gilles Brassard and Paul Bratley,Algorithmics:Theory and Practice,Prentice Hall,1995.

Computer Networks				
Course code	IT304			
Category	Professional Core Course			
Course title	Computer Networks (Theory)			
Scheme and Credits	**Credits**	4+0	Semester –V	

Course Objectives:

- This course studies the standard models for the layered approach to communication between autonomous machines in a network, and the main characteristics of data transmission across various physical link types. It considers how to design networks and protocols for diverse situations, analyses several application and support protocols from a distributed systems viewpoint, and identifies significant problem areas in networked communications.

Unit-1	**Introductory Concepts:** Goals and applications of networks, Categories of networks, Organization of the Internet, ISP, Network structure and architecture (layering principles, services, protocols, and standards), The OSI reference model, TCP/IP protocol suite, Network devices and components. **Physical Layer:** Network topology design, Types of connections, Transmission media, Signal transmission and encoding, Network performance and transmission impairments, Switching techniques and multiplexing.	**12 (Lectures)**
Unit-2	**Link Layer:** Framing, Error Detection and Correction, Flow control (Elementary Data Link Protocols, Sliding Window protocols). Medium Access Control and Local Area Networks: Channel allocation, Multiple access protocols, LAN standards, Link layer switches & bridges (learning bridge and spanning tree algorithms).	**12 (Lectures)**
Unit-3	**Network Layer:** Point-to-point networks, Logical addressing, Basic internetworking (IP, CIDR, ARP, RARP, DHCP, ICMP), Routing, forwarding and delivery, Static and dynamic routing, Routing algorithms and protocols, Congestion control algorithms, IPv6.	**12 (Lectures)**
Unit-4	**Transport Layer:** Process-to-process delivery, Transport layer protocols (UDP and TCP), Multiplexing, Connection management, Flow control and retransmission, Window management, TCP Congestion control, Quality of service. **Application Layer:** Domain Name System, World Wide Web and Hyper Text Transfer Protocol, Electronic mail, File Transfer Protocol, Remote login, Network management, Data compression, Cryptography – basic concepts.	**12 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Explain basic concepts, OSI reference model, services, and role of each layer of OSI model and TCP/IP, networks devices and transmission media, Analog and digital data transmission.

CO2 Apply channel allocation, framing, error, and flow control techniques.

CO3 Describe the functions of Network Layer i.e., Logical addressing, subnetting & Routing Mechanism.

CO4 Explain the different Transport Layer functions i.e., Port addressing, Connection Management, Error control and Flow control mechanism.

CO5 Explain the functions offered by session and presentation layer and their Implementation.

CO6 Explain the different protocols used at application layer i.e., HTTP, SNMP, SMTP, FTP, TELNET and VPN.

Textbooks & References:

1. Behrouz Forouzan, “Data Communication and Networking”, McGraw Hill
2. Andrew Tanenbaum “Computer Networks”, Prentice Hall.
3. William Stallings, “Data and Computer Communication”, Pearson.
4. Kurose and Ross, “Computer Networking- A Top-Down Approach”, Pearson.
5. Peterson and Davie, “Computer Networks: A Systems Approach”, Morgan Kaufmann
6. W. A. Shay, “Understanding Communications and Networks”, Cengage Learning.
7. D. Comer, “Computer Networks and Internets”, Pearson.
8. Behrouz Forouzan, “TCP/IP Protocol Suite”, McGraw Hill.

Internet Programming using Java			
Course code	IT305		
Category	Professional Core Course		
Course title	Internet Programming using Java (Theory)		
Scheme and Credits	**Credits**	3+0	Semester –V

Course Objectives:

- To learn why Java is useful for the design of desktop and web applications.
- To learn how to implement object-oriented designs with Java.
- To identify Java language components and how they work together in applications.
- To design and program stand-alone Java applications.

Unit-1	**Introduction:** Introduction and Web Development Strategies, History of Web and Internet, Protocols governing Web, Writing Web Projects, Connecting to Internet, Introduction to Internet services and tools, Introduction to client-server computing. **Core Java:** Introduction, Operator, Data type, Variable, Arrays, Methods & Classes, Inheritance, Package and Interface, Exception Handling, Multithread programming, I/O, Java Applet, String handling, Event handling, Introduction to AWT, AWT controls, Layout managers.	**9 (Lectures)**
Unit-2	**Web Page Designing & Scripting** **HTML:** list, table, images, frames, forms, CSS, Document type definition. **XML:** DTD, XML schemes, Object Models, presenting and using XML, Using XML **Processors:** DOM and SAX, Dynamic HTML. **Java script:** Introduction, documents, forms, statements, functions, objects; introduction to AJAX, VB Script, Introduction to Java Beans, Advantage, Properties, BDK, Introduction to EJB, Java Beans API.	**9 (Lectures)**
Unit-3	**Server Site Programming:** Introduction to active server pages (ASP), Introduction to Java Server Page (JSP), JSP Application Design, JSP objects, Conditional Processing, declaring variables and methods, sharing data between JSP pages, Sharing Session and Application Data, Database Programming using JDBC, development of java beans in JSP, Introduction to Servlets, Lifecycle, JSDK, Servlet API, Servlet Packages, Introduction to COM/DCOM/CORBA.	**9 (Lectures)**
Unit-4	**PHP (Hypertext Preprocessor):** Introduction, syntax, variables, strings, operators, if-else, loop, switch, array, function, form, mail, file upload, session, error, exception, filter, PHP-ODBC.	**9 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Introduce the fundamentals of Java.

CO2Construct basic websites using HTML and Scripting.

CO3 Introduce the fundamentals of Server Site Programming.

CO4 Develop modern interactive web applications using PHP, XML and MySQL

Textbooks & References:

1. Burdman, Jessica, "Collaborative Web Development" Addison Wesley
2. Xavier, C, " Web Technology and Design" , New Age International
3. Ivan Bayross," HTML, DHTML, Java Script, Perl & CGI", BPB Publication

4. Bhave, “Programming with Java”, Pearson Education
5. Herbert Schieldt, “The Complete Reference:Java”, TMH.
6. Hans Bergsten, “Java Server Pages”, SPD O’Reilly 6. Ullman, “PHP for the Web: Visual QuickStart Guide”, Pearson Education
7. Margaret Levine Young, “The Complete Reference Internet”, TMH 8. Naughton, Schildt, “The Complete Reference JAVA2”, TMH 9. Balagurusamy E, “Programming in JAVA”, TMH
8. Ramesh Bangia, “Internet and Web Design” , New Age International
9. Ivan Bayross,” HTML, DHTML, Java Script, Perl & CGI”, BPB Publication
10. Deitel, “Java for programmers”, Pearson Education
11. Chris Bates, “Web Programing Building Internet Applications”, 2nd Edition, WILEY, Dreamtech
12. Joel Sklar , “Principal of web Design” Vikash and Thomas Learning 6. Horstmann, “CoreJava”, Addison Wesley

Database Management System Lab			
Course code	IT352		
Category	Professional Core Course		
Course title	Database Management System Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –V

EXP-1 Installing oracle/ MYSQL
EXP-2 Creating Entity-Relationship Diagram using case tools.
EXP-3 Writing SQL statements Using ORACLE /MYSQL:

a. Writing basic SQL SELECT statements.
b. Restricting and sorting data.
c. Displaying data from multiple tables.
d. Aggregating data using group function.
e. Manipulating data.
f. Creating and managing tables.

EXP-4 Implementing Normalization
EXP-5 Creating cursor
EXP-6 Creating procedure and functions
EXP-7 Creating packages and triggers
EXP-8 Design and implementation of payroll processing system
EXP-9 Design and implementation of Library Information System
EXP-10 Design and implementation of Student Information System
EXP-11 Automatic Backup of Files and Recovery of Files

Sample of Mini project (Design & Development of Data and Application) for following:

a) Inventory Control System.
b) Material Requirement Processing.
c) Hospital Management System.
d) Railway Reservation System.
e) Personal Information System.
f) Web Based User Identification System.
g) Timetable Management System.
h) Hotel Management System

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Design and Analysis of Algorithm Lab			
Course code	IT353		
Category	Professional Core Course		
Course title	Design and Analysis of Algorithm Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –V

EXP-1 To analyze time complexity of Insertion Sort, Merge Sort and Quick Sort.
EXP-2 To Implement Strassen's Matrix Multiplication.
EXP-3 To implement Merge Sort using Divide and Conquer approach.
EXP-4 To implement Quick Sort using Divide and Conquer approach.
EXP-5 To implement Knapsack Problem.
EXP-6 To implement Activity Selection Problem
EXP-7 To implement Dijkstra's Algorithm.
EXP-8 To implement Bellman Ford's Prim's.
EXP-9 To implement Kruskal's Algorithms.
EXP-10 To implement Largest Common Subsequence.
EXP-11 To implement Matrix Chain Multiplication.
EXP-12 To implement Multistage Graph Algorithms.
EXP-13 To implement n-Queen Algorithms.
EXP-14 To implement Naïve String-Matching Algorithm.
EXP-15 To implement Rabin Karp String Matching Algorithm.

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Computer Network Lab			
Course code	IT354		
Category	Professional Core Course		
Course title	Computer Network Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –V

EXP-1 Implementation of Stop and Wait Protocol and Sliding Window Protocol.
EXP-2 Study of Socket Programming and Client – Server model.
EXP-3 Write a code simulating ARP /RARP protocols.
EXP-4 Write a code simulating PING and TRACEROUTE commands.
EXP-5 Create a socket for HTTP for web page upload and download.
EXP-6 Write a program to implement RPC (Remote Procedure Call).
EXP-7 Implementation of Subnetting.
EXP-8 Applications using TCP Sockets like
a) Echo client and echo server
b) Chat
c) File Transfer
EXP-9 Applications using TCP and UDP Sockets like
a) DNS
b) SNMP
c) File Transfer
EXP-10 Study of Network simulator (NS).and Simulation of Congestion Control Algorithms using NS
EXP-11 Perform a case study about the different routing algorithms to select the network path with its optimum and economical during data transfer.
a) Link State routing
b) Flooding
c) Distance vector
EXP-12 To learn handling and configuration of networking hardware like RJ-45 connector, CAT-6 cable, crimping tool, etc.
EXP-13 Configuration of router, hub, switch etc. (using real devices or simulators)
EXP-14 Running and using services/commands like ping, traceroute, nslookup, arp, telnet, ftp, etc.
EXP-15 Network packet analysis using tools like Wireshark, tcpdump, etc.
EXP-16 Network simulation using tools like Cisco Packet Tracer, NetSim, OMNeT++, NS2, NS3, etc.
EXP-17 Socket programming using UDP and TCP (e.g., simple DNS, data & time client/server, echo client/server, iterative &concurrent servers)

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Internet Programming using Java Lab			
Course code	IT355		
Category	Professional Core Course		
Course title	Internet Programming using Java Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –V

EXP-1 Create a web page with the following using HTML:
- a) To embed a map in a web page.
- b) To fix the hot spots in that map.
- c) Show all the related information when the hot spots are clicked.

EXP-2 Create a web page with the following:
- a) Cascading style sheets.
- b) Embedded style sheets.
- c) Inline style sheets.

EXP-3 Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.

EXP-4 Write programs in Java using Servlets:
- a) To invoke servlets from HTML forms.
- b) Session tracking using hidden form fields and Session tracking for a hit count.

EXP-5 Write programs in Java to create three-tier applications using servlets for conducting online examination for displaying student mark list. Assume that student information is available in a database which has been stored in a database server.

EXP-6 Install TOMCAT web server. Convert the static web pages of programs into dynamic web pages using servlets (or JSP) and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.

EXP-7 Redo the previous task using JSP by converting the static web pages into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database.

EXP-8 Create and save an XML document at the server, which contains 10 users Information. Write a Program, which takes user Id as an input and returns the User details by taking the user information from the XML document.

EXP-9 Validate the form using PHP regular expression. ii. PHP stores a form data into database.

EXP-10 Write a web service for finding what people think by asking 500 people‘s opinion.

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Principle of Compiler Design				
Course code	IT306			
Category	Professional Core Course			
Course title	Principle of Compiler Design (Theory)			
Scheme and Credits	**Credits**	3+0	Semester –VI	

Course Objectives:

The aim of this course is to provide students with the knowledge and abilities to design and implement compilers.

Unit-1	**Compiler Structure:** Analysis-Synthesis Model of Compilation, Various Phases of a Compiler, Tool Based Approach to Compiler Construction Lexical Analysis: Interface with Input, Parser and Symbol Table, Token, Lexeme and Patterns, Difficulties in Lexical Analysis, Error Reporting, and Implementation. Regular Definition, Transition Diagrams, LEX.	**9 (Lectures)**
Unit-2	**Syntax Analysis:** Context Free Grammars, Ambiguity, Associativity, Precedence, Top-Down Parsing, Recursive Descent Parsing, Transformation on the Grammars, Predictive Parsing, Bottom-Up Parsing, Operator Precedence Grammars, LR Parsers (SLR, LALR, LR), YACC.	**9 (Lectures)**
Unit-3	**Syntax Directed Definitions:** Inherited and Synthesized Attributes, Dependency Graph, Evaluation Order, Bottom Up and Top-Down Evaluation of Attributes, L- and S-Attributed Definitions. Type Checking: Type System, Type Expressions, Structural and Name Equivalence of Types, Type Conversion, Overloaded Functions and Operators, Polymorphic Functions. Intermediate Code Generation: Intermediate Representations, Translation of Declarations, Assignments Intermediate Code Generation for Control Flow, Boolean Expressions and Procedure Calls, Implementation Issues	**9 (Lectures)**
Unit-4	Symbol Table Management, Runtime Environments, Source Language Issues, Storage Organization, Storage Allocation Strategies, Access to Non-Local Names, Parameter Passing. Code Optimization, Peephole Optimization, Source of Optimizations, Optimization of Basic Blocks, Loops, Global Dataflow Analysis, Introduction to Code Generation.	**9 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Define the phases of a typical compiler, including the front and back end.

CO2 Explain the role of a parser in a compiler and relate the yield of a parse tree to a grammar derivation; design and implement a parser using a typical parser generator.

CO3 Apply an algorithm for a top-down or a bottom-up parser construction; construct a parser for a small context free grammar.

CO4 Describe the purpose of translating to intermediate code in the compilation process.

CO5 Design and implement an intermediate code generator based on given code patterns.

Textbooks & References:

1. Aho, Sethi & Ullman, "Compilers: Principles, Techniques and Tools", Pearson Education
2. K. Muneeswaran, Compiler Design, First Edition, Oxford University Press.
3. J.P. Bennet, "Introduction to Compiler Techniques", Second Edition, McGraw-Hill,2003.
4. Henk Alblas and Albert Nymeyer, "Practice and Principles of Compiler Building with C", PHI, 2001.
5. V Raghvan, "Principles of Compiler Design", McGraw-Hill.
6. Kenneth Louden," Compiler Construction", Cengage Learning.
7. Charles Fischer and Ricard LeBlanc," Crafting a Compiler with C", Pearson Education.Agents, Cambridge University Press, 2010.

Machine Learning Techniques				
Course code	IT307			
Category	Professional Core Course			
Course title	Machine Learning Techniques (Theory)			
Scheme and Credits	**Credits**	4+0	Semester –VI	

Course Objectives:

- To understand the basic theory underlying machine learning.
- To be able to formulate machine learning problems corresponding to different applications.
- To understand a range of machine learning algorithms along with their strengths and weaknesses.
- To be able to apply machine learning algorithms to solve problems of moderate complexity.

To apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models.

Unit-1	**Foundations of Learning:** Components of Learning, Learning Models, Geometric Models, Probabilistic Models, Logic Models, Grouping and Grading, Learning Versus Design, Types of Learning, Supervised, Unsupervised, Reinforcement, Theory of Learning, Feasibility of Learning, Error and Noise, Training versus Testing, Theory of Generalization, Generalization Bound, Approximation, Generalization Tradeoff, Bias and Variance, Learning Curve.	**12 (Lectures)**
Unit-2	**Linear Models and Linear Classification:** Univariate Linear Regression, Multivariate Linear Regression, Regularized Regression, Logistic Regression, Perceptron, Multilayer Neural Networks, Learning Neural Networks Structures, Support Vector Machines, Soft Margin SVM, Going Beyond Linearity, Generalization and Over Fitting, Regularization, Validation.	**12 (Lectures)**
Unit-3	**Distance-Based Models and Tree:** Nearest Neighbour Models– K-Means Clustering around Medoids, Silhouettes, Hierarchical Clustering, K-D Trees, Locality Sensitive Hashing, Non-Parametric Regression, Ensemble Learning, Bagging and Random Forests, Boosting, Meta Learning, Decision Trees, Learning Decision Trees, Ranking and Probability Estimation Trees, Regression Trees, Clustering Trees.	**12 (Lectures)**
Unit-4	**Rule Models and Reinforcement Learning:** Learning Ordered Rule Lists, Learning Unordered Rule Lists, Descriptive Rule Learning, Association Rule Mining, First Order Rule Learning, Passive Reinforcement Learning, Direct Utility Estimation, Adaptive Dynamic Programming, Temporal-Difference Learning, Active Reinforcement Learning, Exploration, Learning an Action-Utility Function, Generalization in Reinforcement Learning, Policy Search, Applications in Game Playing, Applications in Robot Control.	**12 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Explain theory underlying machine learning.
CO2 Construct algorithms to learn linear and non-linear models.
CO3 Implement data clustering algorithms.
CO4 Construct algorithms to learn tree and rule-based models.
CO5 Apply reinforcement learning techniques.

Textbooks & References:

1. Ethem Alpaydın -Introduction to Machine Learning Third Edition, MIT Press, 2004.
2. Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, Learning from Data, AML Book Publishers, 2012.
3. P. Flach, Machine Learning: The art and science of algorithms that make sense of data,

Cambridge University Press, 2012.
4. K. P. Murphy, Machine Learning: A probabilistic perspective, MIT Press, 2012.
5. C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2007.
6. D. Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012.
7. M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning, MIT Press, 2012.
8. T. M. Mitchell, Machine Learning, McGraw Hill, 1997.
9. S. Russel and P. Norvig, Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall, 2009.

Advanced Topics in Data Mining				
Course code	IT308			
Category	Professional Core Course			
Course title	Advanced Topics in Data Mining (Theory)			
Scheme and Credits	**Credits**	3+0	Semester –VI	

Course Objectives:

- To develop knowledge of algorithms for massive data sets and methodologies in the context of data mining.
- To gain experience in matching various algorithms for particular classes of problems.
- To gain experience in applying and developing algorithms as a part of software development for mining big data.

Unit-1	**Data Warehousing and Business Analysis:** Data warehousing Components, Building a Data warehouse, Data Warehouse Architecture, DBMS Schemas for Decision Support, Data Extraction, Cleanup, and Transformation Tools, Metadata, reporting, Query Tools and Applications, Online Analytical Processing (OLAP) and Multidimensional Data Analysis.	**6 (Lectures)**
Unit-2	**Data Mining:** Data Mining Functionalities, Data Preprocessing, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization and Concept Hierarchy Generation, Architecture of A Typical Data Mining Systems, Classification of Data Mining Systems. **Association Rule Mining:** Efficient and Scalable Frequent Item set Mining Methods, Mining Various Kinds of Association Rules, Association Mining to Correlation Analysis, Constraint-Based Association Mining.	**9 (Lectures)**
Unit-3	**Classification and Prediction:** Issues Regarding Classification and Prediction, Classification by Decision Tree Introduction, Bayesian Classification, Rule Based Classification, Classification by Back propagation, Support Vector Machines, Associative Classification, Lazy Learners, Other Classification Methods, Prediction, Accuracy and Error Measures, Evaluating the Accuracy of a Classifier or Predictor, Ensemble Methods, Model Section.	**8 (Lectures)**
Unit-4	**Cluster Analysis:** Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning Methods, Hierarchical methods, Density-Based Methods, Grid-Based Methods, Model-Based Clustering Methods, Clustering High-Dimensional Data, Constraint-Based Cluster Analysis, Outlier Analysis. **Mining Object, Spatial, Multimedia, Text and Web Data:** Multidimensional Analysis and Descriptive Mining of Complex Data Objects, Spatial Data Mining, Multimedia Data Mining, Text Mining, Mining the World Wide Web.	**12 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Understand what Is Data Mining, what kinds of data can be mined, what kinds of patterns can be mined, and what kinds of applications are targeted.

CO2 Explain major Issues in data mining.

CO3 Apply machine learning, pattern recognition, statistics, visualization, algorithm, database technology and high-performance computing in data mining applications.

CO4 Identify what kinds of technologies are used for different applications.

CO5 Manipulate data preprocessing, data Warehouse and OLAP technology, data cube technology; mining frequent patterns and association, classification, clustering, and outlier detection.

Textbooks & References:

1. Alex Berson and Stephen J. Smith "Data Warehousing, Data Mining & OLAP", Tata McGraw – Hill Edition, Tenth Reprint 2007.
2. K.P. Soman, Shyam Diwakar and V. Ajay "Insight into Data mining Theory and Practice", Easter Economy Edition, Prentice Hall of India, 2006.
3. G. K. Gupta "Introduction to Data Mining with Case Studies", Easter Economy Edition, Prentice Hall of India, 2006.
4. Pang-Ning Tan, Michael Steinbach and Vipin Kumar "Introduction to Data Mining", Pearson Education, 2007.

Principle of Compiler Design Lab			
Course code	IT356		
Category	Professional Core Course		
Course title	Principle of Compiler Design Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –VI

EXP-1 Design and implement a lexical analyzer for given language using C and the lexical analyzer should ignore redundant spaces, tabs and new lines.

EXP-2 Implementation of Lexical Analyzer using Lex Tool

a) Generate YACC specification for a few syntactic categories.
b) Program to recognize a valid arithmetic expression that uses operator +, – , * and /.
c) Program to recognize a valid variable which starts with a letter followed by any number of letters or digits.
d) Implementation of Calculator using LEX and YACC
e) Convert the BNF rules into YACC form and write code to generate abstract syntax tree.

EXP-3 Write program to find ε – closure of all states of any given NFA with ε transition.

EXP-4 Write program to convert NFA with ε transition to NFA without ε transition.

EXP-5 Write program to convert NFA to DFA

EXP-6 Write program to minimize any given DFA.

EXP-7 Develop an operator precedence parser for a given language.

EXP-8 Write program to find Simulate First and Follow of any given grammar.

EXP-9 Construct a recursive descent parser for an expression.

EXP-10 Construct a Shift Reduce Parser for a given language.

EXP-11 Write a program to perform loop unrolling.

EXP-12 Write a program to perform constant propagation.

EXP-13 Implement Intermediate code generation for simple expressions.

EXP-14 Implement the back end of the compiler which takes the three-address code and produces the 8086 assembly language instructions that can be assembled and run using an 8086 assembler. The target assembly instructions can be simple move, add, sub, jump etc.

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Machine Learning Techniques Lab			
Course code	IT357		
Category	Professional Core Course		
Course title	Machine Learning Techniques Lab (Laboratory)		
Scheme and Credits	**Credits**	0+1	Semester –VI

EXP-1 Write a python program to import and export data using Pandas library functions
EXP-2 Demonstrate various data pre-processing techniques for a given dataset 6
EXP-3 Implement Dimensionality reduction using Principle Component Analysis (PCA) method.
EXP-4 Write a Python program to demonstrate various Data Visualization Techniques.
EXP-5 Implement Simple and Multiple Linear Regression Models.
EXP-6 Develop Logistic Regression Model for a given dataset.
EXP-7 Develop Decision Tree Classification model for a given dataset and use it to classify a new sample.
EXP-8 Implement Naïve Bayes Classification in Python
EXP-9 Build KNN Classification model for a given dataset.
EXP-10 Build Artificial Neural Network model with back propagation on a given dataset.
a) Implement Random Forest ensemble method on a given dataset.
b) Implement Boosting ensemble method on a given dataset.
EXP-11 Write a python program to implement K-Means clustering Algorithm.

Note: Instructor may add/delete/modify/tune experiments, wherever he/she feels in a justified manner.

Department of Information Technology,
Institute of Engineering and Technology, Deen Dayal Upadhyaya Gorakhpur University, Gorakhpur

IT Professional Elective Course

Detailed Syllabus

Neural Networks			
Course code	ITE115		
Category	Professional Elective Course		
Course title	Neural Networks (Theory)		
Scheme and Credits	**Credits**	3+0	

Course Objectives:

- To enable students to understand important concepts and theories of neural networks.
- To enable students to understand how artificial neural networks can be designed and trained.
- To enable students to calculate simple examples of neural networks.
- To give students an appreciation of some of the limitations and possibilities of neural networks.

Unit-1	**Fundamentals of Neural Networks:** A Neural Network, Human Brain, Models of a Neuron, Neural Networks viewed as Directed Graphs, Network Architectures, Knowledge Representation, Artificial Intelligence and Neural Networks, Properties of Different Learning Rules, Types of Activation Functions, Training of Artificial Neural Network, Perceptron Model (Both Single &Multi-Layer), Training Algorithm, Problems Solving Using Learning Rules and Algorithms, Linear Separability Limitation and Its Over Comings.	**9 (Lectures)**
Unit-2	**Single Layer Perceptions:** Adaptive Filtering Problem, Unconstrained Organization Techniques, Linear Least Square Filters, Least Mean Square Algorithm, Learning Curves, Learning Rate Annealing Techniques. **Multi-Layer Networks:** Back Propagation Networks (BPN), Training of BPN, Architecture and Algorithm of BPN, Counter Propagation Network (CPN), Training of CPN, Architecture CPN, Bi-Directional Associative Memory (BAM), Training-stability analysis of BAM, Adaptive Resonance Theory (ART), - ART1- ART2 Architecture and Training of ART, Hop Field Network.	**10 (Lectures)**
Unit-3	**Self-Organization Maps (SOM):** Two Basic Feature Mapping Models, Self-Organization Map, SOM Algorithm, Properties of Feature Map, Computer Simulations, Learning Vector Quantization, Adaptive Patter Classification, Linear vector quantization, Probabilistic neural network, General Regression neural network, Application of Artificial Neural Network, Texture classification - Character recognition.	**8 (Lectures)**
Unit-4	**Introduction to Fuzzy Logic:** Classical Set, Operations and properties of Classical Set, Fuzzy Set, Operations and properties of Fuzzy Set, Classical Relations, Operations and Properties of Classical Relations, Fuzzy Relations, Operations and Properties of Fuzzy Relations, Compositions Membership function, Fuzzy Logic Controller (FLC), Need for FLC, Fuzzification, Defuzzification.	**9 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 understand the principles of Neural Networks.
CO2 Perform the training of Artificial Neural Networks using various learning rules.
CO3 Perform the testing of Neural Networks.
CO4 Perform analysis of these networks for various pattern recognition applications.
CO5 Identify different types of models of Neural Networks.

Textbooks & References:

1. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016.
2. Bishop, C. ,M., Pattern Recognition and Machine Learning, Springer, 2006.
3. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
4. Golub, G.,H., and Van Loan,C.,F., Matrix Computations, JHU Press,2013.
5. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.
6. Simon Haykin, "Artificial Neural Networks", Second Edition, Pearson Education.
7. Laurene Fausett, "Fundamentals of Neural Networks, Architectures, Algorithms and Applications", Prentice Hall publications.
8. Neural Networks in Computer Intelligence, Li Min Fu MC GRAW HILL EDUCATION 2003.
9. Neural Networks -James A Freeman David M S Kapura Pearson Education 2004.
10. Introduction to Artificial Neural Systems Jacek M. Zurada, JAICO Publishing House Ed. 2006.

Computer graphics and Multimedia			
Course code	ITE116		
Category	Professional Elective Course		
Course title	Computer graphics and Multimedia (Theory)		
Scheme and Credits	**Credits**	3+0	

Course Objectives:

- To make the students understand graphics concepts and develop, design and implement 2D and 3D graphical structures.
- To understand multimedia compression techniques and applications of multimedia.

Unit-1	**Basics Of Computer Graphics**: Introduction, Area of Computer Graphics, Design and Drawing, Animation Multimedia Applications, Simulation, Methods for Storing and Displaying Pictures, Difficulties for Displaying Pictures. **Graphic Devices:** Cathode Ray Tube, Quality of Phosphors, CRTs for Color Display, Beam Penetration CRT, Shadow Mask CRT, Direct View Storage Tube, Tablets, Light Pen, Three Dimensional Devices. **Simple Line Drawing Methods**: Point Plotting Techniques, Qualities of Good Line Drawing Algorithms, Digital Differential Analyzer (DDA) Line Drawing Algorithms, Bresenham's Line Drawing Algorithm, Mid-Point Circle Generation Algorithm, Bresenham's Circle Generation Algorithm.	**9 (Lectures)**
Unit-2	**Two Dimensional Transformations, Clipping and Windowing:** Definition of Transformation, Matrix Representation of Points, Basic Transformation, Need for Clipping and Windowing, Line Clipping Algorithms, Midpoint Subdivision Method, Other Clipping Methods, Sutherland - Hodgeman Algorithm, Viewing Transformations. **Graphical Input Techniques:** Graphical Input Techniques, Positioning Techniques, Positional Constraints, Rubber Band Techniques. **Event Handling and Input Functions:** Introduction, Polling, Event Queue, Functions for Handling Events, Polling Task Design, Input Functions, Dragging and Fixing, Hit Detection, OCR.	**10 (Lectures)**
Unit-3	**Three-Dimensional Graphics:** Need for 3-Dimensional Imaging, Techniques for 3- Dimensional Displaying, Parallel Projections, Perspective Projection, Intensity Cues, Stereoscope Effect, Kinetic Depth Effect, Shading. **Curves And Surfaces:** Shape Description Requirements, Parametric Functions, Bezier Methods, Bezier Curves, Bezier Surfaces, B-Spline Methods.	**9 (Lectures)**
Unit-4	**Solid Area Scan Conversion:** Three Dimensional Transformations Solid Area Scan Conversion, Scan Conversion of Polygons, Algorithm Singularity, Three-Dimensional Transformation, Translations, Scaling, Rotation, Viewing Transformation, Perspective, Algorithms, Three-Dimensional Clipping, Perspective View of Cube. **Hidden Surface Removal:** Need for Hidden Surface Removal, Depth - Buffer Algorithm, Properties that Help in Reducing Efforts.	**8 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Understand the graphics hardware used in field of computer graphics.

CO2 Understand the concept of graphics primitives such as lines and circle based on different algorithms.

CO3 Apply the 2D graphics transformations, composite transformation and Clipping concepts.

CO4 Apply the concepts and techniques used in 3D computer graphics, including viewing

transformations.

CO5 Perform the concept of projections, curve and hidden surfaces in real life.

Textbooks & References:

1. Donald Hearn and M Pauline Baker, "Computer Graphics C Version", Pearson Education.
2. Foley, Vandam, Feiner, Hughes – "Computer Graphics principle", Pearson Education.
3. Rogers, "Procedural Elements of Computer Graphics", McGraw Hill.
4. W. M. Newman, R. F. Sproull – "Principles of Interactive computer Graphics" – McGraw Hill.
5. Amrendra N Sinha and Arun D Udai," Computer Graphics", McGraw Hill.
6. R.K. Maurya, "Computer Graphics" Wiley Dreamtech Publication.
7. Mukherjee, Fundamentals of Computer graphics & Multimedia, PHI Learning Private Limited.
8. Donald Hearn and M Pauline Baker, "Computer Graphics with Open GL", Pearson education.

Mobile Ad-hoc Network			
Course code	ITE117		
Category	Professional Elective Course		
Course title	Mobile Ad-hoc Network (Theory)		
Scheme and Credits	**Credits**	3+0	

Course Objectives:

This course covers major aspects of ad hoc networks, from design through performance issues to application requirements. It starts with characteristics features, applications of ad hoc networks, Modulation techniques and voice coding. It also covers the IEEE Wireless LAN and Bluetooth standards.

Unit-1	**Introduction:** Introduction to ad-hoc networks, definition, characteristics features, applications, Characteristics of wireless channel, ad-hoc mobility models, indoor and outdoor models.	**9 (Lectures)**
Unit-2	**MAC Protocols:** Design issues, goals and classification. Contention based protocols– with reservation, scheduling algorithms, protocols using directional antennas. IEEE standards: 802.11a, 802.11b, 802.11g, 802.15. HIPERLAN.	**10 (Lectures)**
Unit-3	**Routing Protocols:** Design issues, goals and classification, Proactive Vs reactive routing, unicast routing algorithms, Multicast routing algorithms, hybrid routing algorithm, energy aware routing algorithm, hierarchical routing, QoS aware routing.	**9 (Lectures)**
Unit-4	**Transport Layer:** Issues in designing, Transport layer classification, ad-hoc transport protocols. Security issues in ad-hoc networks: issues and challenges, network security attacks, secure routing protocols, **Cross layer Design:** Need for cross layer design, cross layer optimization, parameter optimization techniques, cross layer cautionary perspective. Integration of ad-hoc with Mobile IP networks.	**8 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Explain basic concepts, OSI reference model, services and role of each layer of OSI model and TCP/IP, networks devices and transmission media, Analog and digital data transmission.

CO2 Apply channel allocation, framing, error and flow control techniques.

CO3 Describe the functions of Network Layer i.e., Logical addressing, subnetting & Routing Mechanism.

CO4 Explain the different Transport Layer function i.e., Port addressing, Connection Management, Error control and Flow control mechanism.

CO5 Explain the different protocols used at application layer i.e., HTTP, SNMP, SMTP, FTP, TELNET and VPN.

Textbooks & References:

1. C. Siva Ram Murthy and B. S. Manoj, Ad hoc Wireless Networks Architecture and Protocols, 2nd edition, Pearson Edition, 2007.
2. Charles E. Perkins, Ad hoc Networking, Addison – Wesley, 2000.
3. Stefano Basagni, Marco Conti, Silvia Giordano and Ivan stojmenovic, Mobile ad-hoc networking, Wiley-IEEE press, 2004.
4. Mohammad Ilyas, The handbook of ad-hoc wireless networks, CRC press, 2002.
5. T. Camp, J. Boleng, and V. Davies " A Survey of Mobility Models for Ad-hoc Network".
6. Research, "Wireless Commun, and Mobile Comp.. Special Issue on Mobile Ad-hoc Networking Research, Trends and Applications, Vol. 2, no. 5, 2002, pp. 483 – 502.
7. A survey of integrating IP mobility protocols and Mobile Ad-hoc networks, Fekri M. bduljalil and Shrikant K. Bodhe, IEEE communication Survey and tutorials, no: 12007

Deep Learning			
Course code	ITE215		
Category	Professional Elective Course		
Course title	Deep Learning (Theory)		
Scheme and Credits	**Credits**	3+0	

Course Objectives:

- To introduce the idea of artificial neural networks and their architecture.
- To introduce techniques used for training artificial neural networks.
- To enable design of an artificial neural network for classification.
- To enable design and deployment of deep learning models for machine learning problems.

Unit-1	**Introduction:** Feedforward Neural networks. Gradient descent and the backpropagation algorithm. Unit saturation, aka the vanishing gradient problem, and ways to mitigate it. RelU Heuristics for avoiding bad local minima. Heuristics for faster training. Nestors accelerated gradient descent. Regularization. Dropout.	**6 (Lectures)**
Unit-2	**Convolutional Neural Networks:** Architectures, convolution/pooling layers. **Recurrent Neural Networks:** Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Encoder Decoder Architectures. **Deep Unsupervised Learning**: Autoencoders, Variational Autoencoders, Adversarial Generative Networks, Autoencoder and Deep Boltzmann Machine (DBM).	**10 (Lectures)**
Unit-3	**Applications of Deep Learning to Computer Vision:** Image segmentation, object detection, automatic image captioning, Image generation with Generative adversarial networks, video to text with LSTM models. Attention models for computer vision tasks, Attention and memory models, Dynamic memory networks.	**8 (Lectures)**
Unit-4	**Applications of Deep Learning to Natural Language Processing (NLP):** Introduction to NLP and Vector Space Model of Semantics, Continuous Skip-Gram Model, Continuous Bag-of Words Model (CBOW), Glove, Evaluations and Applications in Word Similarity, Analogy Reasoning Named Entity Recognition, Parsing and Sentiment Analysis using Recursive Neural Networks, Sentence Classification using Convolutional Neural Networks, Applications of Dynamic Memory Networks in NLP. **Recent Research in NLP using Deep Learning:** Factoid Question Answering, similar question detection, Dialogue topic tracking, Neural Summarization, Smart Reply.	**12 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Able to understand the mathematics behind functioning of artificial neural networks.
CO2 Able to analyze the given dataset for designing a neural network-based solution.
CO3 Able to carry out design and implementation of deep learning models for signal/image processing applications.
CO4 Able to design and deploy simple TensorFlow-based deep learning solutions to classification problems.

Textbooks & References:

1. Bengio, Yoshua, Ian J. Goodfellow, and Aaron Courville. "Deep learning." An MIT Press book in preparation. (2015).

2. Bengio, Yoshua. "Learning deep architectures for AI." Foundations and trends in Machine Learning 2.1 (2009): 1127.
3. Hochreiter, Sepp, and Jargen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 17351780.

Mobile Computing			
Course code	ITE216		
Category	Professional Elective Course		
Course title	Mobile Computing (Theory)		
Scheme and Credits	**Credits**	3+0	

Course Objectives:

- To understand the basic concepts of mobile computing.
- To learn the basics of mobile telecommunication systems.
- To be familiar with the network layer protocols and Ad-Hoc networks.
- To know the basis of transport and application layer protocols.

To gain knowledge about different mobile platforms and application development.

Unit-1	**Introduction:** Issues in Mobile Computing, Overview of Wireless Telephony: Cellular Concept, GSM: Air-Interface, Channel Structure, Location Management: HLR, VLR, Hierarchical, Handoffs, Channel Allocation in Cellular Systems, CDMA, GPRS.	**9 (Lectures)**
Unit-2	**Wireless Networking:** Wireless LAN Overview: MAC Issues, IEEE 802.11, Blue Tooth, Wireless Multiple Access Protocols, TCP Over Wireless, Wireless Applications, Mobile IP. **WAP:** Architecture, Protocol Stack, Application Environment, Applications, Wireless mark Up Language (WML).	**10 (Lectures)**
Unit-3	Data Management Issues, Data Replication for Mobile Computers, Adaptive Clustering for Mobile Wireless Networks, File System, Disconnected Operations, Mobile Agents Computing, Security and Fault Tolerance.	**9 (Lectures)**
Unit-4	**Ad-Hoc Networks:** Localization, MAC Issues, Routing Protocols, Global State Routing (GSR), Destination Sequenced Distance Vector Routing (DSDV), Dynamic Source Routing (DSR), Ad Hoc On Demand Distance Vector Routing (AODV), Temporary Ordered Routing Algorithm (TORA), QOS in Ad Hoc Network.	**8 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Demonstrate the actual meaning of power and energy management in wireless mobile networks.

CO2 Be familiar with the network protocol stack

CO3 Learn the basics of mobile telecommunication system

CO4 Be exposed to Ad-Hoc networks and Mobile IP

CO5 Gain knowledge about different mobile platforms and application development

Textbooks & References:

1. Asoke K Taukder, Roopa R Yavagal, Mobile Computing, Tata McGraw Hill Pub. Co., New Delhi, 2005.
2. J. Schiller, Mobile Communication, Addison Wesley, 2000.
3. Ivan Stojmenovic, Handbook of Wireless Networks and Mobile Computing, John Wiley & sons Inc, Canada, 2002.
4. William Stallings, "Wireless Communication and Networks", Pearson Education, 2003.
5. Yi-Bing Lin & Imrich Chlamtac, Wireless and Mobile Networks Architectures, John Wiley & Sons, 2001.
6. Raj Pandya, "Mobile and Personal Communication systems and services", Prentice Hall of India, 2001.
7. Hansmann, "Principles of Mobile Computing", Wiley Dreamtech, 2004.
8. Ray Rischpater, "Wireless Web Development", Springer Publishing, 2000.
9. P. Stavronlakis, "Third Generation Mobile Telecommunication systems", Springer Publishers,

2001.
10. Burkhardt, Pervasive Computing, Pearson
11. P. Stavronlakis, Third Generation Mobile Telecommunication systems, Springer Publishers.

IT Open Elective Courses

Detailed Syllabus

Introduction to OOP with C++			
Course code	ITOE01		
Category	Engineering Open Elective Course		
Course title	Introduction to OOP with C++ (Theory)		
Scheme and Credits	**Credits**	3+0	
Course Objectives: The objective of course is to develop programming skills of students, using object-oriented programming concepts, learn the concept of class and object using C++ and develop classes for simple applications.			
Unit-1	**Introduction to Object Oriented Programming:** Basic concept of OOP, Comparison of Procedural Programming and OOP, Benefits of OOP, C++ compilation, Abstraction, Encapsulation, Inheritance, Polymorphism, Difference between C and C++. **Elements of C++ Language:** Tokens and identifiers: Character set and symbols, Keywords, C++ identifiers; Variables and Constants: Integer, character and symbolic constants; Dynamic initialization of variables, Reference variables, Basic data types in C++, Streams in C++. **Operators and Manipulators:** Operators, Types of operators in C++, Precedence and associativity of operators, Manipulators.		**9 (Lectures)**
Unit-2	**Decision and Control:** Structures if statement, if-else statement, switch statement, Loop: while, do-while, for; Jump statements: break, continue, go to. **Functions:** main () function, components of function: prototype, function call, definition, parameter; passing arguments; types of function, inline function, function overloading. **Array, Pointer and Structure:** Arrays, pointers, structures, unions. **Introduction to Classes and Objects:** Classes in C++, class declaration, declaring objects, Defining Member functions, Inline member function, Array of objects, Objects as function argument, Static data member and member function, Friend function and friend class.		**9 (Lectures)**
Unit-3	**Constructors and Destructors:** Constructors, Instantiation of objects, Default constructor, Parameterized constructor, Copy constructor and its use, Destructors, Constraints on constructors and destructors, Dynamic initialization of objects. **Operator Overloading:** Overloading unary operators: Operator keyword, arguments and return value; overloading unary and binary operators: arithmetic operators, manipulation of strings using operators, Type conversions.		**9 (Lectures)**
Unit-4	**Inheritance:** Derived class and base class: Defining a derived class, Accessing the base class member, Inheritance: multilevel, multiple, hierarchical, hybrid; Virtual base class, Abstract class. **Virtual Functions and Polymorphism:** Virtual functions, pure virtual functions; Polymorphism, Categorization of polymorphism techniques: Compile time polymorphism, Run time polymorphism. **File Handling:** File classes, Opening and Closing a file, File modes, Manipulation of file pointers, Functions for I/O operations.		**9 (Lectures)**
Course Outcomes (CO) At the end of course, student will be able to			

CO1 Identify importance of object-oriented programming and difference between structured oriented and object-oriented programming features.
CO2 Make use of objects and classes for developing programs.
CO3 Understand, analyze and apply the role of overall modeling concepts (i.e., System, structural)
CO4 Understand, analyze and apply oops concepts (i.e. abstraction, inheritance, polymorphism)
CO5 Use various object-oriented concepts to solve different problems.

Textbooks & References:

1. James Rumbaugh et. al, "Object Oriented Modeling and Design", Pearson Education.
2. Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide", Pearson Education.
3. Object Oriented Programming with C++, E Balagurusamy, McGraw Hill.
4. C++ Programming, Black Book, Steven Holzner, dreamtech.
5. Object Oriented Programming in Turbo C++, Robert Lafore, Galgotia.
6. Object Oriented Programming with ANSI and Turbo C++, Ashok Kamthane, Pearson
7. The Compete Reference C++, Herbert Schlitz, McGraw Hill.

Introduction to Virtualization and Cloud Computing			
Course code	ITOE02		
Category	Engineering Open Elective Course		
Course title	Introduction to Virtualization and Cloud Computing (Theory)		
Scheme and Credits	**Credits**	3+0	

Course Objectives:

- To give the students introductory information about current practices in virtualization and cloud computing.
- To explain virtualized operating systems, their installation and implementation.
- To introduce computing models, techniques and architectures.
- To provide practical knowledge on designing and implementing virtual and cloud-based software systems and major providers of such systems in the market today.
- To introduce use in enterprise level information management.

Unit-1	**Introduction To Virtualization:** Traditional IT Infrastructure, Benefits of Virtualization, Types of Virtualizations, History of Virtualization, Server, Storage, Network and Application Virtualization: Types of Server Virtualization, Hypervisors, Anatomy of Server Virtualization, Benefits of Storage Virtualization, Types of Storage Virtualization, VPN, VLAN, Benefits of Application Virtualization.	**9 (Lectures)**
Unit-2	**Introduction to Cloud Computing:** History, Importance of Virtualization in Cloud, Anatomy of Cloud, Cloud deployment models, Cloud delivery models, Steppingstones for the development of cloud, Grid Computing, Cloud Computing.	**9 (Lectures)**
Unit-3	**Cloud Implementations:** Decision Factors for Cloud Implementations, Public, Private and Hybrid Cloud, Overview, Infrastructure as a Service (IaaS) Cloud Delivery Model, Platform as a Service (PaaS) Cloud Delivery Model, Software as a Service (SaaS) Cloud Delivery Model.	**9 (Lectures)**
Unit-4	**Case Study on Virtualization, Cloud Workloads:** Customer IT Landscape, Triggers of Virtualization, Preparation for Virtualization, Transition Tools for Virtualization, Cost savings, Cloud workload Overview, Workloads most suitable for Cloud, Workloads not suitable for Cloud.	**9 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Significance and requirement of Cloud Computing.
CO2 Analyze use of Hypervisor and Its Types.
CO3 Understand Anatomy and Delivery Model of Cloud.
CO4 Cloud Workload Management.

Textbooks & References:

1. Cloud Computing: Fundamentals, Industry Approach and Trends, by Rishab Sharma, Wiley Publication
2. Mastering Cloud Computing, by Rajkumar Buyya.
3. Kai Hwang, Geoffrey C. Fox and Jack J. Dongarra, “Distributed and Cloud Computing from Parallel Processing to the Internet of Things”, Morgan Kaufmann, Elsevier, 2012.
4. Barrie Sosinsky, “Cloud Computing Bible” John Wiley & Sons, 2010.
5. Tim Mather, Subra Kumaraswamy, and Shahed Latif, “Cloud Security and Privacy An Enterprise Perspective on Risks and Compliance”, O'Reilly 2009.

Cyber Law and Ethics			
Course code	ITOE03		
Category	Engineering Open Elective Course		
Course title	Cyber Law and Ethics (Theory)		
Scheme and Credits	**Credits**	3+0	

Course Objectives:

- To explain the authorities applicable to given cyber operations scenario.
- To provide a high-level explanation of the legal issues governing the authorized conduct of cyber operations and the use of related tools, techniques, technology, and data.
- To evaluate the relationship between ethics and law, describe civil disobedience and its relation to ethical hacking, describe criminal penalties related to unethical hacking.
- To describe steps for carrying out ethical penetration testing, describe 'ethical hacking' principles and conditions, distinguish between ethical and unethical hacking, and distinguish between nuisance hacking, activist hacking, criminal hacking, and acts of war.

Unit-1	**Introduction to Cyber Law:** Evolution of computer technology, emergence of cyber space, Cyber Jurisprudence, Jurisprudence and law, Doctrinal approach, Consensual approach, Real Approach, Cyber Ethics, Cyber Jurisdiction, Hierarchy of courts, Civil and criminal jurisdictions, Cyberspace-Web space, Web hosting and web Development agreement, Legal and Technological Significance of domain Names, Internet as a tool for global access.	**9 (Lectures)**
Unit-2	**Information Technology Act:** Overview of IT Act, 2000, Amendments and Limitations of IT Act, Digital Signatures, Cryptographic Algorithm, Public Cryptography, Private Cryptography, Electronic Governance, Legal Recognition of Electronic Records, Legal Recognition of Digital Signature, Certifying Authorities, Cyber Crime and Offences, Network Service Providers Liability, Cyber Regulations Appellate Tribunal, Penalties and Adjudication.	**9 (Lectures)**
Unit-3	**Cyber Law and Related Legislation:** Patent Law, Trademark Law, Copyright, Software – Copyright or Patented, Domain Names and Copyright disputes, Electronic Data Base and its Protection, IT Act and Civil Procedure Code, IT Act and Criminal Procedural Code, Relevant Sections of Indian Evidence Act, Relevant Sections of Bankers Book Evidence Act, Relevant Sections of Indian Penal Code, Relevant Sections of Reserve Bank of India Act, Law Relating To Employees And Internet, Alternative Dispute Resolution , Online Dispute Resolution (ODR).	**9 (Lectures)**
Unit-4	**Electronic Business and Legal Issues:** Evolution and development in E-commerce, paper vs paper less contracts E-Commerce models- B2B, B2C, E security. Business, taxation, electronic payments, supply chain, EDI, E-markets, Emerging Trends, Cyber Ethics: The Importance of Cyber Law, Significance of cyber-Ethics, Need for Cyber regulations and Ethics. Ethics in Information society, Introduction to Artificial Intelligence Ethics: Ethical Issues in AI and core Principles, Introduction to Block chain Ethics.	**9 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Understand cyber laws.
CO2 Learn basics of cyber laws and ethics.
CO3 Authentication and security measures.
CO4 Determine appropriate mechanisms for protecting from cyber-crime.
CO5 Design a security solution for a given application, system with respect to security of the system.

Textbooks & References:

1. "Investigating Cyber Law and Cyber Ethics: Issues, Impacts and Practices" by Alfreda Dudley and James Braman.
2. "Cyber Law: A Legal Arsenal for Online Business" by Brett Trout.
3. "Cybersecurity Ethics: An Introduction" by Mary Manjikian.
4. "Computer Law and Ethics (Computer Science)" by Charles Thies.
5. Cyber War: Law and Ethics for Virtual Conflicts" by Ohlin.

Internet of Things			
Course code	ITOE04		
Category	Engineering Open Elective Course		
Course title	Internet of Things (Theory)		
Scheme and Credits	**Credits**	3+0	

Course Objectives:

- Recognize the factors that contributed to the emergence of IoT.
- Design and program IoT devices.
- Use real IoT protocols for communication.
- Secure the elements of an IoT device.
- Define the infrastructure for supporting IoT deployments.

Unit-1	**Fundamentals Of IoT:** Evolution of Internet of Things, Enabling Technologies. **IoT Architectures:** oneM2M, IoT World Forum (IoTWF) and Alternative IoT models, Simplified IoT Architecture and Core IoT Functional Stack, Fog, Edge and Cloud in IoT, Functional blocks of an IoT ecosystem, Sensors, Actuators, Smart Objects and Connecting Smart Objects.	**9 (Lectures)**
Unit-2	**IoT Protocols:** IoT Access Technologies, Physical and MAC layers, topology and Security of IEEE 802.15.4, 802.15.4g, 802.15.4e, 1901.2a, 802.11ah and LoRaWAN, Network Layer: IP versions, Constrained Nodes and Constrained Networks. **Optimizing IP for IoT:** From 6LoWPAN to 6Lo, Routing over Low Power and Lossy Networks – Application Transport Methods: Supervisory Control and Data Acquisition. **Application Layer Protocols:** CoAP and MQTT.	**9 (Lectures)**
Unit-3	**Design and Development:** Design Methodology, Embedded computing logic, Microcontroller, System on Chips, IoT system building blocks, Arduino, Board details, IDE programming, Raspberry Pi, Interfaces and Raspberry Pi with Python Programming. **Data Analytics and Supporting Services:** Structured Vs Unstructured Data and Data in Motion Vs Data in Rest, Role of Machine Learning, No SQL Databases, Hadoop Ecosystem, Apache Kafka, Apache Spark, Edge Streaming Analytics and Network Analytics, Xively Cloud for IoT, Python Web Application Framework, Django, AWS for IoT, System Management with NETCONF-YANG.	**9 (Lectures)**
Unit-4	**Case Studies/Industrial Applications:** Cisco IoT system, IBM Watson IoT platform, Manufacturing, Converged Plantwide Ethernet Model (CPwE), Power Utility Industry, Grid Blocks Reference Model. **Smart and Connected Cities:** Layered architecture, Smart Lighting, Smart Parking Architecture and Smart Traffic Control.	**9 (Lectures)**

Course Outcomes (CO)

At the end of course, student will be able to

CO1 Explain the concept of IoT.
CO2 Analyze various protocols for IoT.
CO3 Design a PoC of an IoT system using Rasperry Pi/Arduino.
CO4 Apply data analytics and use cloud offerings related to IoT.
CO5 Analyze applications of IoT in real time scenario.

Textbooks & References:

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, —IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, Cisco

Press, 2017.
2. Arshdeep Bahga, Vijay Madisetti, —Internet of Things – A hands-on approach, Universities Press, 2015
3. Olivier Hersent, David Boswarthick, Omar Elloumi, —The Internet of Things – Key applications and Protocols, Wiley, 2012.
4. Jan Ho¨ ller, Vlasios Tsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand. David Boyle, "From Machine-to-Machine to the Internet of Things – Introduction to a New Age of Intelligence", Elsevier, 2014.
5. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), —Architecting the Internet of Things, Springer, 2011.
6. Michael Margolis, Arduino Cookbook, Recipes to Begin, Expand, and Enhance Your Projects, 2nd Edition, O'Reilly Media, 2011.