FACULTY OF ENGINEERING

Scheme of Instruction & Examination

(AICTE Model Curriculum for the Academic Year 2020-2021)

and

Syllabi

B.E. III and IV Semester

of

Four Year Degree Programme

in

Computer Engineering

(With effect from the academic year 2020–2021)
(As approved in the faculty meeting held on **-**-2020)

Issued by

Dean, Faculty of Engineering

Osmania University, Hyderabad – 500 007

2020

SCHEME OF INSTRUCTION & EXAMINATION
B.E. (Computer Engineering) III – SEMESTER

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	P/D	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	HS204ME	Operations Research	3	-	-	3	30	70	3	3
2	BS206BZ	Biology for Engineers	3	-	-	3	30	70	3	3
3	BS205MT	Mathematical foundations for Data Science (Probability & Statistics)	3	-	-	3	30	70	3	3
4	ES214EC	Basic Electronics Engineering	3	-	-	3	30	70	3	3
5	ES216CM	Logic and Switching Theory	3	-	-	3	30	70	3	3
6	PC221CM	Data Structures	3	-	-	3	30	70	3	3
7	PC222CM	Discrete Structure & Mathematical Logic	3	-	-	3	30	70	3	3
Practical/ Laboratory Courses										
8	ES251EC	Basic Electronics Engineering Lab	-	-	2	2	25	50	3	1
9	PC252CM	Data Structures Lab	-	-	2	2	25	50	3	1
10	PC253CM	IT Workshop Lab	-	-	2	2	25	50	3	1
			21	**-**	**06**	**27**	**285**	**640**		**24**

HS: Humanities and Social Sciences BS: Basic Science ES: Engineering Science
MC: Mandatory Course PC: Professional Core
L: Lecture T: Tutorial P: Practical D: Drawing
CIE: Continuous Internal Evaluation SEE: Semester End Evaluation (Univ. Exam)
PY: Philosophy, BZ: Biology/ Life Sciences, CE: Civil Engineering, CS: Computer Science and Engineering
EC: Electronics and Communication Engineering, ME: Mechanical Engineering.

Note:

1. Each contact hour is a clock hour.
2. The duration of the practical class is two hours, however it can be extended wherever necessary, to enable the student to complete the experiment.

Course Code	Course Title						Core/Elective
HS204ME	**Operations Research**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

- Use variables for formulating complex mathematical models in management science, industrial engineering and transportation models.
- Use the basic methodology for the solution of linear programming problems.
- Understand the mathematical tools that are needed to solve optimization problems like Transportation models and Assignment models.
- Understand the replacement models with change in money value considering with time and without time.
- Model a system as a queuing model and compute important performance measures

Course Outcomes

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

1. Prepare the students to have the knowledge of Linear Programming Problem in Operations
2. Research at the end students would be able to understand the concept and develop the models for different applications.
3. Make students understand the concept Replacement models at the end students would able to explain various features and applications of replacement models in real time scenario.
4. Prepare the students to understand theory of Game in operations research at the end students would able to explain application of Game theory in decision making for a conflict
5. Prepare the students to have the knowledge of Sequencing model at the end student would able to develop optimum model for job scheduling.
6. Prepare students to understand Queuing theory concepts and various optimization techniques at the end students would able to develop models for waiting line cases.

UNIT-I

Introduction: Definition and Scope of Operations Research.

Linear Programming: Introduction, Formulation of linear programming problems, graphical method of solving LP problem, simplex method, maximization and minimization, Degeneracy in LPP, Unbounded and, Infeasible solutions.

UNIT-II

Duality: Definition, Relationship between primal and dual solutions, Economic Interpretation, Post optimal of sensitivity analysis, Dual Simplex Method.

UNIT-III

Transportation Models: Finding an initial feasible solution - North West corner method, least cost method, Vogel’s Approximation method, Finding the optimal solution, optimal solution by stepping stone and MODI methods, Special cases in Transportation problems - Unbalanced Transportation problem.

Assignment Problems: Hungarian method of Assignment problem, Maximization in Assignment problem, unbalanced problem, problems with restrictions, travelling salesman problems.

UNIT-IV

Replacement Models: Introduction, replacement of items that deteriorate ignoring change in money value, replacement of items that deteriorate considering change in money value with time, replacement of items that fail suddenly - Individual replacement policy, Group replacement policy.

Game Theory: Introduction, 2 person zero sum games, Maximin - Minimax principle, Principle of Dominance, Solution for mixed strategy problems, Graphical method for 2 x n and m x 2 games.

UNIT-V

Sequencing Models: Introduction, General assumptions, processing n jobs through 2 machines, processing 'n' jobs through m machines, Processing 2 jobs through m machines

Queuing Theory: Introduction, single channel - Poisson arrivals - exponential service times with infinite population & finite population, Multi-channel - Poisson arrivals - Exponential service times with infinite population.

Introduction to Optimization Techniques: Single objective & Multi objective optimization Techniques like G.A, NSGA, P.Q.O & MPSO Techniques.

Suggested Readings:

1. Hamdy, A. Taha, Operations Research-An Introduction, Sixth Edition, Prentice Hall of India Pvt. Ltd., 1997.
2. S.D. Sharma, Operations Research, Kedarnath, Ramnath & Co., Meerut, 2009.
3. Hrvey M. Wagner, Principles of Operations Research, Second Edition, Prentice Hall of India Ltd., 1980.
4. V.K. Kapoor, Operations Research, S. Chand Publishers, New Delhi, 2004.
5. R. Paneer Selvam, Operations Research, Second Edition, PHI Learning Pvt. Ltd., New Delhi, 2008.
6. Data Reconciliation by Prof. Shanker Narasimha

Course Code	Course Title						Core/Elective
BS206BZ	**Biology for Engineers**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

Gain vivid knowledge in the fundamentals and uses of biology, human system and plant system.

Course Outcomes

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

1. Apply biological engineering principles, procedures needed to solve real-world problems.
2. Understand the fundamentals of living things, their classification, cell structure and biochemical constituents.
3. Apply the concept of plant, animal and microbial systems and growth in real life situations.
4. Comprehend genetics and the immune system.
5. Know the cause, symptoms, diagnosis and treatment of common diseases.
6. Apply basic knowledge of the applications of biological systems in relevant industries.

UNIT-I

Introduction to Life: Characteristics of living organisms, Basic classification, cell theory, structure of prokaryotic and eukaryotic cell, Introduction to Biomolecules: definition, general classification and important functions of carbohydrates, lipids, proteins, vitamins and enzymes.

UNIT-II

Biodiversity: Plant System: basic concepts of plant growth, nutrition, photosynthesis and nitrogen fixation. Animal System: Elementary study of digestive, respiratory, circulatory, excretory systems and their functions. Microbial System: History, types of microbes, economic importance and control of microbes.

UNIT-III

Genetics and Evolution: Theories of evolution and Evidences; cell division–mitosis and meiosis; evidence of laws of inheritance; variation and speciation; nucleic acids as a genetic material; central dogma; Mendel laws, gene and chromosomes.

UNIT-IV

Human Diseases: Definition, causes, symptoms, diagnosis, treatment and prevention of diabetes, cancer, hypertension, influenza, AIDS and Hepatitis. Immunity immunization, antigen – antibody immune response

UNIT-V

Biology and its Industrial Applications: Transgenic plants and animals, stem cell and tissue engineering, bioreactors, bio pharming, recombinant vaccines, cloning, drug discovery, biological neural networks, bioremediation, biofertilizer, biocontrol, biofilters, biosensors, biopolymers, bioenergy, biomaterials, biochips, basic biomedical instrumentation.

Suggested Readings:

1. A Text book of Biotechnology, R.C.Dubey, S. Chand Higher Academic Publications, 2013
2. Diseases of the Human Body, Carol D. Tamparo and Marcia A. Lewis, F.A. Davis Company, 2011.
3. Biomedical instrumentation, Technology and applications, R. Khandpur, McGraw Hill Professional, 2004
4. Biology for Engineers, Arthur T. Johnson, CRC Press, Taylor and Francis, 2011
5. Cell Biology and Genetics (Biology: The unity and diversity of life Volume I), Cecie Starr, Ralph Taggart, Christine Evers and Lisa Starr, Cengage Learning, 2008
6. Biotechnology Expanding horizon, B.D. Singh, Kalyani Publishers, 2012.

Course Code	Course Title						Core/Elective
BS205MT	**Mathematical Foundations for Data Science (Probability & Statistics)**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

- To introduce the solution methodologies for second order Partial Differential Equations with applications in engineering
- To provide an overview of probability and statistics to engineers

Course Outcomes

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

1. Solve field problems in engineering involving PDEs.
2. They can also formulate and solve problems involving random variables and apply statistical methods for analysing experimental data.

UNIT-I:Introduction of Probability, Conditional probability, Theorem of Total probability, Baye's Theorem and its applications, Random variables, Types of random variables, Probability mass function and Probability density function, Mathematical expectations.

UNIT-II: Discrete probability distributions: Binomial and Poisson distributions, Mean, variance, moment generating function and evaluation of statistical parameters for these distributions, Moments, Skewness and Kurtosis.

UNIT-III: Continuous probability distributions, Uniform, Exponential and Normal distributions, Mean, variance, moment generating function and evaluation of statistical parameters for these distributions

UNIT-IV: Curve fitting by the method of least squares: Fitting of straight lines, second degree parabolas and more general curves, Correlation, regression and Rank correlation. Test of significance: Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations.

UNIT-V: Test for single mean, difference of means and correlation coefficients, test for ratio of variances, Chi-square test for goodness of fit and independence of attributes.

Suggested Readings:

1. R.K.Jain & Iyengar, "Advanced Engineering Mathematics", Narosa Publications.
2. B.S. Grewal, "Higher Engineering Mathematics", Khanna Publishers, 2000.
3. P.Sivaramakrishna Das & C.Vijaya Kumar, "Engineering Mathematics" , Pearson India Education Services Pvt. Ltd.
4. N.P. Bali & M. Goyal, "A Text Book of Engineering Mathematics", Laxmi Publications, 2010.
5. S.C.Gupta & V.K.Kapoor, "Fundamentals of Mathematical Statistics" , S.Chand Pub.
6. P. G. Hoel, S. C. Port & C. J. Stone, "Introduction to Probability Theory", Universal Book Stall, 2003.
7. W. Feller, "An Introduction to Probability Theory and its Applications", Vol. 1, Wiley, 1968.

<table>
<tr><td>Course Code</td><td colspan="6">Course Title</td><td>Core/Elective</td></tr>
<tr><td><b>ES214EC</b></td><td colspan="6"><b>Basic Electronics Engineering</b></td><td><b>Core</b></td></tr>
<tr><td rowspan="2">Prerequisite</td><td colspan="4">Contact Hours per Week</td><td rowspan="2">CIE</td><td rowspan="2">SEE</td><td rowspan="2">Credits</td></tr>
<tr><td>L</td><td>T</td><td>D</td><td>P</td></tr>
<tr><td><b>-</b></td><td><b>3</b></td><td><b>-</b></td><td><b>-</b></td><td><b>-</b></td><td><b>30</b></td><td><b>70</b></td><td><b>3</b></td></tr>
</table>

Course Objectives

The objectives of this course is to impart knowledge

- To analyze the behavior of semiconductor diodes in Forward and Reverse bias.
- To design of Half wave and Full wave rectifiers with L,C, LC & CLC Filters.
- To explore V-I characteristics of Bipolar Junction Transistor in CB, CE & CC configurations.
- To explain feedback concept and different oscillators.
- To analyze Digital logic basics and Photo Electric devices.

Course Outcomes

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

1. Able to learn about forward biased and reversed biased circuits.
2. Able to plot the V-I Characteristics of diode and transmission.
3. Able to design combinational logic circuits and PLDs.

UNIT-I

Semi-Conductor Theory: Energy Levels, Intrinsic and Extrinsic Semiconductors, Mobility, Diffusion and Drift current. Hall Effect, Characteristics of P-N Junction diode, Parameters and Applications.

Rectifiers: Half wave and Full wave Rectifiers (Bridge, center tapped) with and without filters, ripple regulation and efficiency. Zener diode regulator.

UNIT-II

Bipolar Junction Transistor: BJT, Current components, CE, CB, CC configurations, characteristics, Transistor as amplifier. Analysis of CE, CB, CC Amplifiers (qualitative treatment only) . JFET: Construction and working, parameters, CS, CG, CD Characteristics, CS amplifier.

UNIT-III

Feedback Concepts – Properties of Negative Feedback Amplifiers, Classification, Parameters. Oscillators – Barkhausen Criterion, LC Type and RC Type Oscillators and Crystal Oscillators. (Qualitative treatment only).

UNIT-IV

Operational Amplifiers – Introduction to OP Amp, characteristics and applications –Inverting and Non-inverting Amplifiers, Summer, Integrator, Differentiator, Instrumentation Amplifier. Digital Systems: Basic Logic Gates, half, Full Adder and Subtractors.

UNIT-V

Data Acquisition Systems: Study of transducer (LVDT, Strain gauge, Temperature, and Force). Photo Electric Devices and Industrial Devices: Photo diode, Photo Transistor, LED, LCD, SCR, UJT Construction and Characteristics only.

Display Systems: Constructional details of C.R.O and Applications.

Suggested Readings:

1. Jocob Millman, Christos C. Halkias and Satyabrata Jit,Electronics Devices and Circuits, 3rd Edition, McGraw Hill Education (India) Private Limited, 2010.
2. Rama Kanth A. Gaykward,Op-AMPS and Linear Integrated Circuit, 4th Edition PrenticeHall of India, 2000.
3. M. Morris Mano,Digital Design, 3rd Edition, Prentice Hall of India, 2002.
4. William D Cooper, and A.D. Helfrick, Electronic Measurements and Instrumentations Techniques, 2nd Edition, Prentice Hall of India, 2008.
5. S. Shalivahan, N. Suresh Kumar, A. Vallava Raj,Electronic Devices and Circuits, 2nd Edition., McGraw Hill Education (India) Private Limited, 2007.

Course Code	Course Title						Core/Elective
ES216EC	**Logic and Switching Theory**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives:

- To introduce concepts of Boolean logic, Postulates and Boolean Theorems.
- To understand the use of logic minimization methods and to solve the Boolean logic expressions
- To understand how to design the combinational and sequential circuits. • To introduce and realize the adder circuits
- To understand the state reduction methods for sequential circuits.

Course Outcomes:

Students will be

- Able to apply the concepts of Boolean logic, Postulates and Boolean Theorems to solve the Boolean expressions.
- Able to solve the Complex Boolean logic expressions using Minimization methods.
- Able to design the combinational, sequential circuits and Various adder circuits.
- Able to apply state reduction methods to solve sequential circuits.

UNIT-I

Boolean Algebra: Axiomatic definition of Boolean Algebra Operators, Postulates and Theorems, Boolean Functions, Canonical Forms and Standard Forms, Simplification of Boolean Functions Using Theorems and Karnaugh Map Method.

UNIT-II

Minimization of Switching Functions: Quine-McCluskey Tabular Method, Determination of Prime Implicants and Essential Prime Implicants. Combinational Logic Design: Single-Output and Multiple-Output

Combinational Circuit Design: AND-OR, OR-AND and NAND/NOR Realizations, Exclusive-OR and Equivalence functions.

UNIT–III

Design of Combinational Logic Circuits: Gate Level design of Small Scale Integration (SSI) circuits, Modular Combinational Logic Elements- Decoders, Encoders, Priority encoders, Multiplexers and De-multiplexers.

Design of Integer Arithmetic Circuits using Combinational Logic: Integer Adders – Binary Adders, Subtractors, Ripple Carry Adder and Carry Look Ahead Adder, and Carry Save Adders.

UNIT-IV

Design of Combinational Circuits using Programmable Logic Devices (PLDs): Programmable Read Only Memories (PROMs), Programmable Logic Arrays (PLAs), Programmable Array Logic (PAL) devices.

Introduction to Sequential Circuit Elements: Latch, Various types of Flip-Flops and their Excitation Tables.

UNIT -V

Models of Sequential Circuits: Moore Machine and Mealy Machine, Analysis of Sequential Circuits-State Table and State Transition Diagrams. Design of Sequential Circuits-Counters. Moore and Mealy State Graphs for Sequence Detection, Methods for Reduction of State Tables and State Assignments.

Suggested Reading:

1. M Morris Mano and Michael D Ciletti, Digital Design, Prentice Hall of India, Fourth Edition, 2008.
2. Zvi Kohavi, Switching and Finite Automata Theory, Tata McGraw Hill, 2nd Edition, 1979.
3. R. P Jain, Modern Digital Electronics,4th ed., McGraw Hill Education (India) Private Limited, 2003.
4. Ronald J.Tocci, Neal S. Widmer &Gregory L.Moss, "Digital Systems: Principles and Applications," PHI, 10/e, 2009.
5. Samir Palnitkar, "Verilog HDL A Guide to Digital Design and Synthesis," 2nd Edition, Pearson Education, 2006.

Course Code	Course Title						Core/Elective
PC221CM	**Data Structures**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

- To teach the importance of structuring the data for easy access and storage.
- To teach the implementation of various data structures.
- To acquire skills in using generic principles for data representation and manipulation with a view for efficiency, maintainability and code reuse.
- To introduce the basic concepts of advanced data structures.

Course Outcomes

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

1. Understand the importance of abstract data type and implementing the concepts of data structure using abstract data type.
2. Evaluate an algorithm by using algorithmic performance and measures.
3. Distinguish between linear and non-linear data structures and their representations in the memory using array and linked list.
4. Develop applications using Linear and Non-linear data structures.
5. Apply the suitable data structure for a real world problem and think critically for improvement in solutions.
6. Determine the suitability of the standard algorithms: Searching, Sorting and Traversals.

UNIT-I

Algorithms: Introduction, Algorithm Specifications, Recursive Algorithms, Performance Analysis of an algorithm- Time and Space Complexity, Asymptotic Notations, Complexity Analysis Examples.

Stacks and Queues: ADT Stack and its operations: Algorithms and their complexity analysis,

Applications of Stacks: Expression Conversion and evaluation –corresponding algorithms and complexity analysis.

Queue ADT and its operations: Linear Queue, Circular Queue, Algorithms and their analysis.

UNIT-II

Linked Lists: Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes,

Doubly linked list: Operations on it and algorithmic analysis; Circular Linked Lists, Doubly Linked Lists, Applications (Polynomial Arithmetic).

Arrays and Matrices: Row And Column Major Representations, Sparse Matrices. Hashing: Hash Table Representation, Application- Text Compression.

UNIT- III

Trees: Definitions and Properties, Representation of Binary Trees, Operations, Binary Tree Traversal. Binary

Search Trees: Definitions, Operations on Binary Search Trees.

Balanced Search Trees: AVL Trees, Red Black Trees and B-Trees, Tree operations on each of the trees and their algorithms.

UNIT –IV

Graphs: Definitions and Properties, Representation, Graph Search Methods (Depth First Search and Breadth First Search)

Application of Graphs: Shortest Path Algorithm (Dijkstra), Minimum Spanning Tree (Prim's and Kruskal's Algorithms).

UNIT-V

Sorting and Searching: Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort, Linear and Binary Search algorithms.

Suggested Readings:

1. "Fundamentals of Data Structures in C++", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Dinesh Mehta, 2nd Edition, Universities Press.
2. Algorithms, Data Structures, and Problem Solving with C++", Illustrated Edition by Mark Allen Weiss, 3rd Edition, Pearson India.
3. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.
4. "How to Solve it by Computer", 2nd Impression by R.G. Dromey, Pearson Education.
5. Michael T. Goodrich, Roberto Tamassia, David M. Mount, Data Structures and Algorithms in C++, John Wiley & Sons, 2010.

Course Code	Course Title						Core/Elective
PC222CM	**Discrete Structures & Mathematical logic**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

- To Learn mathematical concepts, terminology and notation as applied in computer science for solving logical problems.
- To Construct correct direct and indirect proofs.
- To Use division into cases in a proof.
- To Use counterexamples.
- Apply logical reasoning to solve a variety of problems
- To model relationships, analyse data, apply probability concepts and use functions to solve problems.
- To develop the mathematical skills needed for advanced quantitative courses.

Course Outcomes

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

1. For a given logic sentence express it in terms of predicates, quantifiers, and logical connectives
2. For a given a problem, derive the solution using deductive logic and prove the solution based on logical inference.
3. For a given a mathematical problem, classify its algebraic structure.
4. Evaluate Boolean functions and simplify expressions using the properties of Boolean algebra.
5. Develop the given problem as graph networks and solve with techniques of graph theory.

UNIT -I

Sets, Relation and Function: Operations and Laws of Sets, Cartesian Products, Binary Relation, Partial Ordering Relation, Equivalence Relation, Image of a Set, Sum and Product of Functions, Bijective functions, Inverse and Composite Function, Size of a Set, Finite and infinite Sets, Countable and uncountable Sets, Cantor's diagonal argument and The Power Set theorem, Schroeder-Bernstein theorem. **Principles of Mathematical Induction:** The Well-Ordering Principle, Recursive definition, The Division algorithm: Prime Numbers, The Greatest Common Divisor: Euclidean Algorithm, The Fundamental Theorem of Arithmetic.

UNIT-II

Basic counting techniques-inclusion and exclusion, pigeon-hole principle, permutation and combination.

UNIT-III

Propositional Logic: Syntax, Semantics, Validity and Satisfiability, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Rules of Inference, The use of Quantifiers. **Proof Techniques:** Some Terminology, Proof Methods and Strategies, Forward Proof, Proof by Contradiction, Proof by Contraposition, Proof of Necessity and Sufficiency.

UNIT-IV

Algebraic Structures and Morphism: Algebraic Structures with one Binary Operation, Semi Groups, Monoids, Groups, Congruence Relation and Quotient Structures, Free and Cyclic Monoids and Groups, Permutation Groups, Substructures, Normal Subgroups, Algebraic Structures with two Binary Operation, Rings, Integral Domain and Fields. Boolean Algebra and Boolean Ring, Identities of Boolean Algebra, Duality, Representation of Boolean Function, Disjunctive and Conjunctive Normal Form

UNIT–V

Graphs and Trees: Graphs and their properties, Degree, Connectivity, Path, Cycle, Sub Graph, Isomorphism, Eulerian and Hamiltonian Walks, Graph Colouring, Colouring maps and Planar Graphs, Colouring Vertices, Colouring Edges, List Colouring, Perfect Graph, definition properties and Example, rooted trees, trees and sorting, weighted trees and prefix codes, Bi-connected component and Articulation Points, Shortest distances.

Suggested books :

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw – Hill.

2. Susanna S. Epp, Discrete Mathematics with Applications,4th edition, Wadsworth Publishing Co. Inc.

3. C L Liu and D P Mohapatra, Elements of Discrete Mathematics A Computer Oriented Approach, 3rd Edition by, Tata McGraw – Hill.

Suggested reference books:

1. J.P. Tremblay and R. Manohar, Discrete Mathematical Structure and It's Application to Computer Science", TMG Edition, TataMcgraw-Hill

2. Norman L. Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press. Schaum's Outlines Series, Seymour Lipschutz, Marc Lipson,

3. Discrete Mathematics, Tata McGraw - Hill

Course Code	Course Title	Core/Elective
ES251EC	**Basic Electronics Engineering Lab**	**Core**

Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**-**	**-**	**-**	**2**	**25**	**50**	**1**

Course Objectives

- To understand the characteristics of diodes and transistor configurations
- To understand the design concepts of biasing of BJT and FET
- To understand the design concepts of feedback amplifiers and oscillators
- To study the design concepts of OP Amp and data converters

Course Outcomes

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

1. Ability to design diode circuits & understand the application of Zener diode.
2. Ability to analyse characteristics of BJTs & FETs.
3. Ability to understand the different oscillator circuits.
4. Ability to understand operation of HWR & FWR circuits with & without filters.
5. Ability tom design Analog-to-Digital converters & Digital-to-Analog converters.

List of Experiments:

1. CRO-Applications, Measurements of R, L and C using LCR meter, Colour code method and soldering practice.
2. Characteristics of Semiconductors diode (Ge,Si and Zener)
3. Static Characteristics of BJT-Common Emitter
4. Static Characteristics of BJT-Common Base
5. Static Characteristics of FET
6. RC-Phase Shift Oscillator
7. Hartley and Colpitts Oscillators
8. Common Emitter Amplifier
9. Astable Multivibrator
10. Full-wave rectifier with and without filters using BJT
11. Operational Amplifier Applications
12. Strain Gauge Measurement
13. Analog-to-Digital and Digital to Analog Converters

Suggested Readings:

1. Maheshwari and Anand, *Laboratory Experiments and PSPICE Simulations in Analog Electronics,* 1st edition, Prentice Hall of India, 2006.
2. David Bell A., *Laboratory Manual for Electronic Devices and Circuits,* Prentice Hall of India, 2001.

Course Code	Course Title						Core/Elective
PC252CM	**Data Structures Lab**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**-**	**-**	**-**	**2**	**25**	**50**	**1**

Course Objectives

- Design and construct simple programs by using the concepts of structures as abstract data type.
- To have a broad idea about how to use pointers in the implement of data structures.
- To enhance programming skills while improving their practical knowledge in data structures.
- To strengthen the practical ability to apply suitable data structure for real time applications.

Course Outcomes

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

1. Implement the abstract data type and reusability of a particular data structure.
2. Implement linear data structures such as stacks, queues using array and linked list.
3. Understand and implements non-linear data structures such as trees, graphs.
4. Implement various kinds of searching, sorting and traversal techniques and know when to choose which technique.
5. Understanding and implementing hashing techniques.
6. Decide a suitable data structure and algorithm to solve a real world problem.

1. Implementation of Stacks, Queues (using both arrays and linked lists).
2. Implementation of circular queue using arrays.
3. Implementation of double ended queue (de queue) using arrays.
4. Implement a program to evaluate a given postfix expression using stacks.
5. Implement a program to convert a given infix expression to postfix form using stacks.
6. Implement the following operations on singly linked list:
   i) Creation ii) Insertion iii) Deletion iv) Traversal
7. Implementation of Polynomial arithmetic using linked list.
8. Implement the following operations on doubly linked list:
   i) Creation ii) Insertion iii) Deletion iv) Traversal
9. Implement the following operations on circular linked list:
   i) Creation ii) Insertion iii) Deletion iv) Traversal
10. Implementation of recursive and iterative traversals on binary tree.
11. Implementation of operations on binary tree (delete entire tree, copy entire tree, mirror image, level order, search for a node etc.)
12. Implementation of the following operations on binary search tree (BST):
    (a) Minimum key (b) Maximum key (c) Search for a given key (d) Delete a node with given key
13. Implement the following operations on AVL search tree:
    i) Insertion ii) Deletion
14. Implement the following operations on B-Trees:
    i) Creation ii) Insertion iii) Deletion iv) Traversal
15. Implementation of graph traversals by applying: (a) BFS (b) DFS
16. Implement the following algorithms to find out a minimum spanning tree of a simple connected undirected graph: (a) Prim's algorithm (b) Kruskal's algorithm
17. Implement Dijkstra's algorithm for solving single source shortest path problem.
18. Implementation of recursive and non recursive functions to perform the following searching operations for a key value in a given list of integers: i) Linear search ii) Binary search
19. Implement the following sorting algorithms:
    a) Bubble sort b) Selection sort c) Insertion sort (d) Merge sort (e) Quick sort (f) Heap sort
20. Implementation of hashing with (a) Separate Chaining and (b) Open addressing methods.

Course Code	Course Title						Core/Elective
PC253CM	**IT Workshop Lab**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**-**	**-**	**-**	**2**	**25**	**50**	**1**

Course Objectives

- Introducing a new object oriented programming
- Enabling students to learn Big Data, Machine Learning etc.
- Preparing students to cope up with new Market tendencies
- To learn programs in MATLAB environment
- To handle Functions, Polynomials by using MATLAB commands
- Ability to solve any Mathematical functions
- To learn Mathematical Modelling in a new approach
- To plot Graphics (2-D) easily and effectively

Course Outcomes

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

1. Implement basic syntax in python.
2. Analyse and implement different kinds of OOP concept in real world problems.
3. Implement MATLAB operations and graphic functions.

SECTION 1 : MAT LAB / SCILAB PROGRAMS

1. Introduction to MATLAB/SCIlab Environment. MATLAB Menus, Toolbars, Computing with MATLAB, Script Files and the Editor/Debugger, MATLAB help System.

2. MATLAB controls: Relational Logical Variables. Conditional Statements: if – else – elseif, switch. Loops: for – while – break, continue. User-Defined Functions.

3. Arrays, Matrices and Matrix Operations Debugging MATLAB Programs. Working with Data Files, and Graphing Functions: XY Plots – Sub-plots.

4.To solve linear equation

5. Solution of Linear equations for Underdetermined and Over determined cases.

6. Determination of Eigen values and Eigen vectors of a Square matrix.

7. Solution of Difference Equations.

8. Solution of Difference Equations using Euler Method.

9. Solution of differential equation using 4th order Runge- Kutta method.

10. Determination of roots of a polynomial.

11.Determination of polynomial using method of Least Square Curve Fitting.

12.Determination of polynomial fit, analyzing residuals, exponential fit and error bounds from the given data.

13.Determination of time response of an R-L-C circuit

SECTION 2 : Python Programs

1 Introduction to Python Programming:

A. Running instructions in Interactive interpreter and a Python Script.
B. Write a program to purposefully raise Indentation Error and Correct it
C. Write a program to compute distance between two points taking input from the user
D. Write a program add.py that takes 2 numbers as command line arguments and prints its sum.
E. Program to display the following information: Your name, Full Address, Mobile Number, College Name, Course Subjects
F. Write a Program for checking whether the given number is a even number or not.

2 Control Structures, Lists

A. Program to find the largest three integers using if-else

B. Program that receives a series of positive numbers and display the numbers in order and their sum

C. Program to find the product of two matrices and

D. Program to display two random numbers that are to be added, the program should allow the student to enter the answer.

E. If the answer is correct, a message of congratulations should be displayed.

F. If the answer is incorrect, the correct answer should be displayed.

G. Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, . 1/10.

H. Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

3 Functions and Recursion

A. Write recursive and non-recursive functions for the following

B. To find GCD of two integers

C. To find the factorial of positive integer

D. To print Fibonacci Sequence up to given number n

E. To display prime number from 2 to n.

F. Function that accepts two arguments: a list and a number n. It displays all of the numbers in the list that are greater than n

G. Functions that accept a string as an argument and return the number of vowels and consonants that the string contains

4 Files, Exceptions, Lists, Sets, Random Numbers

A. Program to write a series of random numbers in a file from 1 to n and display.

B. Program to write the content in a file and display it with a line number followed by a colon

C. Program to display a list of all unique words in a text file

D. Program to analyse the two text files using set operations

E. Write a program to print each line of a file in reverse order.

F. Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?

G. Write a program combine lists that combines these lists into a dictionary.

5 Object Oriented Programming

A. Program to implement the inheritance

B. Program to implement the polymorphism

6 GUI Programming

A. Program that converts temperature from Celsius to Fahrenheit

B. Program that displays your details when a button is clicked

C. Write a GUI for an Expression Calculator using tk

Suggested Readings:

1. Mark Summerfield, “Programming in Python: A Complete Introduction to the Python Language”, Addison-Wesley Professional, 2009.
2. Martin C. Brown,” PYTHON: The Complete Reference”, McGraw-Hill, 2001.
3. W.J. Palm III, Introduction to MATLAB 7 for Engineers, McGraw-Hill International Edition, 2005.
4. Wesley J Chun,” Core Python Applications Programming”, Prentice Hall, 2012.
5. Allen B Downey,” Think Python”, O’Reilly, 2012.
6. Stormy Attaway, “MATLAB: A Practical Introduction to Programming and Problem Solving”.3rd Edition.

SCHEME OF INSTRUCTION & EXAMINATION
B.E. (Computer Engineering) IV – SEMESTER

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	P/D	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	HS201EG	Effective Technical Communication in English	3	-	-	3	30	70	3	3
2	HS202MC	Finance and Accounting	3	-	-	3	30	70	3	3
3	PC231CM	OOP using JAVA	3	-	-	3	30	70	3	3
4	PC234CM	Operating System Concepts	3	-	-	3	30	70	3	3
5	PC233CM	Database Management Systems	3	1	-	4	30	70	4	3.5
6	PC232CM	Computer Organization & Microprocessor	3	-	-	3	30	70	3	3
Practical/ Laboratory Courses										
7	PC261CM	Computer Organization & Microporcessor Lab	-	-	3	3	25	50	3	1.5
8	PC262CM	OOP using JAVA Lab	-	-	2	2	25	50	2	1
9	PC263CM	Database Management Systems Lab	-	-	2	2	25	50	2	1
10	PC264CM	Operating System Concepts Lab	-	-	2	2	25	50	2	1
			18	**1**	**09**	**28**	**280**	**620**		**23**

HS: Humanities and Social Sciences BS: Basic Science ES: Engineering Science
MC: Mandatory Course PC: Professional Core
L: Lecture T: Tutorial P: Practical D: Drawing
CIE: Continuous Internal Evaluation SEE: Semester End Evaluation (Univ. Exam)
PO: Political Science, EG: English, CM: Commerce, MT: Mathematics,
CS: Computer Science and Engineering, EC: Electronics and Communication Engineering,

Note:

1. Each contact hour is a clock hour
2. The duration of the practical class is two hours, however it can be extended wherever necessary, to enable the student to complete the experiment.
3. The students have to undergo a Summer Internship of two-week duration after IV – Semester and credits will be awarded in V – Semester after evaluation.

Course Code	Course Title						Core/Elective
HS201EG	**Effective Technical Communication in English**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

To expose the students to:

- Features of technical communication
- Types of professional correspondence
- Techniques of report writing
- Basics of manual writing
- Aspects of data transfer and presentations.

Course Outcomes

On successful completion of the course, the students would be able to:

1. Handle technical communication effectively
2. Use different types of professional correspondence
3. Use various techniques of report writing
4. Acquire adequate skills of manual writing
5. Enhance their skills of information transfer and presentations

UNIT-I

Definition and Features of Technical communication: Definition and features of technical communication (precision, relevance, format, style, use of visual aids), Differences between general writing and technical writing, Types of technical communication (oral and written)

UNIT-II

Technical Writing-I (Official correspondence): Emails, IOM, Business letters, Business proposals.

UNIT-III

Technical writing-II (Reports): Project report, Feasibility report, Progress report, Evaluation report.

UNIT-IV

Technical writing- III (Manuals): Types of manuals, User manual, Product manual, Operations manual.

UNIT-V

Information Transfer and Presentations: Non-verbal (bar diagram, flow chart, pie chart, tree diagram) to verbal (writing), Verbal (written) to non-verbal, Important aspects of oral and visual presentations.

Suggested Readings:

1. Raman, Meenakshi & Sharma, Sangeeta. (2015). *Technical Communication: Principles and Practice* (3rd ed.). New Delhi, OUP.
2. Rizvi, Ashraf, M. (2017). *Effective Technical Communication* (2nd ed.). New Delhi, Tata McGraw Hill Education.
3. Sharma, R. C., & Mohan, Krishna. (2017). *Business Correspondence and Report Writing: A Practical Approach to Business & Technical Communication* (4th ed.). New Delhi, Tata McGraw Hill Education.
4. Tyagi, Kavita & Misra, Padma. (2011). *Advanced Technical Communication*. New Delhi, PHI Learning.
5. Jungk, Dale. (2004). *Applied Writing for Technicians*. New York, McGraw-Hill Higher Education.

Course Code	Course Title						Core/Elective
HS202MC	**Finance and Accounting**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

The course will introduce the students

- To provide basic understanding of Financial and Accounting aspects of a business unit
- To provide understanding of the accounting aspects of business
- To provide understanding of financial statements
- To provide the understanding of financial system
- To provide inputs necessary to evaluate the viability of projects
- To provide the skills necessary to analyse the financial statements

Course Outcomes

After successful completion of the course the students will be able to

1. Evaluate the financial performance of the business unit.
2. Take decisions on selection of projects.
3. Take decisions on procurement of finances.
4. Analyse the liquidity, solvency and profitability of the business unit.
5. Evaluate the overall financial functioning of an enterprise.

UNIT-I

Basics of Accounting: Financial Accounting–Definition- Accounting Cycle – Journal - Ledger and Trial Balance-Cash Book-Bank Reconciliation Statement (including Problems)

UNIT-II

Final Accounts: Trading Account-Concept of Gross Profit- Profit and Loss Account-Concept of Net Profit-Balance Sheet (including problems with minor adjustments)

UNIT-III

Financial System and Markets: Financial System-Components-Role-Considerations of the investors and issuers- Role of Financial Intermediaries. Financial Markets-Players- Regulators and instruments - Money Markets Credit Market- Capital Market (Basics only)

UNIT-IV

Basics of Capital Budgeting techniques: Time Value of money- Compounding- Discounting- Future Value of single and multiple flows- Present Value of single and multiple Flows- Present Value of annuities-Financial Appraisal of Projects– Payback Period, ARR- NPV, Benefit Cost Ratio, IRR (simple ratios).

UNIT-V

Financial statement Analysis: Financial Statement Analysis- Importance-Users-Ratio Analysis-liquidity, solvency, turnover and profitability ratios.

Suggested Readings:

1. Satyanarayana. S.V. and Satish. D., Finance and Accounting for Engineering, Pearson Education
2. Rajasekharan, Financial Accounting, Pearson Education
3. Sharma.S.K. and Rachan Sareen, Financial Management, Sultan Chand
4. Jonathan Berk, Fundamentals of Corporate Finance, Pearson Education
5. Sharan, Fundamentals of Financial Management, Pearson Education.

Course Code	Course Title						Core/Elective
PC231CM	**OOP using JAVA**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

- To understand fundamentals of object-oriented programming in Java which includes defining classes, invoking methods, difference between applet and application programs, using class libraries
- To create Java application programs using sound OOP practices such as interfaces, exception handling, multithreading.
- Use Collection framework, AWT and event handling to solve real world problems.
- Exploring Swing, and implementing Servlets.

Course Outcomes

1. Achieve proficiency in object-oriented concepts and also learns to incorporate the same into the Java programming language.
2. Create Java application programs using sound OOP practices e.g. Inheritance, interfaces and proper program structuring by using packages, access control specifiers.
3. Understand and Implement the concepts of Exception Handling in java.
4. Develop the ability to solve real-world problems through software development in high-level programming language using Large APIs of Java as well as the Java standard class library.
5. Understand File, Streams, Input and Output Handling in java.
6. Create graphical user interface and Applets in java as well as apply the knowledge of Event Handling.

UNIT- I

Object Oriented Programming: Principles, Benefits of Object Oriented Programming.

Introduction to Java: Java buzzwords, bytecode. Java Programming Fundamentals: Applet and Application program using simple java program, data types, variables, arrays, operators, expressions, control statements, type conversion and casting, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, introducing access control, static, final, nested and inner classes, exploring string class, using command-line arguments.

Inheritance: Inheritance concept, types of inheritance, Member access rules, use of super and final.

Polymorphism - dynamic binding, method overriding, abstract classes and methods.

UNIT - II

Interfaces: Defining an interface, implementing interfaces, extending interface.

Packages: Defining, Creating and Accessing a Package, importing packages

Exception handling: Benefits of exception handling, classification, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, built in exceptions, creating own exception sub classes.

Multithreading: Java Thread Model, The Main Thread, creating a Thread, creating multiple threads, using is Alive() and join(), thread priorities, synchronization, inter thread communication, deadlock.

UNIT- III

Collections: Overview of Java Collection frame work, commonly used Collection classes – Array List,

Linked List, Hash Set, Tree Set, Collection Interfaces – Collection, List, Set. Accessing Collection via iterator, working with Map. Legacy classes and interfaces – Vector, Hashtable, Stack, Dictionary, Enumeration interface.

Other Utility classes: String Tokenizer, Date, Calendar, Gregorian Calendar, Scanner

Java Input/Output: exploring java.io, Java I/O classes and interfaces, File, Stream classes, byte stream, character stream, serialization.

UNIT- IV

GUI Programming with java: The AWT class hierarchy, MVC architecture. Applet Revisited: Basics, architecture and skeleton, simple applet program.

Event Handling: Delegation Event Model, Event Classes, Source of Events, Event Listener Interfaces. Handling mouse and keyboard events, Adapter classes.

Database Programming using JDBC: Introduction to JDBC, JDBC Drivers & Architecture, CURD operation Using JDBC, Connecting to non-conventional Databases.

UNIT V

Exploring Swing: JLabel, ImageIcon, JTextField, the Swing buttons, JTabbedpane, JScrollPane, JList, JComboBox.

Servlet: Life cycle, using tomcat, simple servlet, servlet API, javax.servlet package, reading servlet parameters, javax.servlet.http package, handling HTTP requests and responses

Suggested Readings:

1. Herbert Scheldt, "The Complete Reference Java, 7th Edition, Tata McGraw Hill, 2006.

2. James M Slack, Programming and Problem Solving with JAVA, Thomson Learning, 2002.

3. C Thomas Wu, An Introduction to Object Oriented Programming with Java 5th Edition, McGraw Hill Publishing, 2010.

4. H. M. Dietel and P. J. Dietel, Java How to Program, Sixth Edition, Pearson Education / PHI.

Suggested Reference Readings:

1. Understanding OOP with Java, up dated edition, T. Budd, Pearson education.
2. Head First Java, 2nd Edition by Bert Bates, Kathy Sierra Publisher: O'Reilly Media, Inc.
3. An Introduction to programming and OO design using Java, J. Nino and F.A. Hosch, John Wiley & sons.
4. An Introduction to OOP, second edition, T. Budd, Pearson Education.
5. Introduction to Java programming 6th edition, Y. Daniel Liang, Pearson Education.
6. An introduction to Java programming and object oriented application development, R. A. Johnson-Thomas.

Course Code	Course Title						Core/Elective
PC232CM	**Computer Organization & Microprocessor**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

- To understand basic components of computers.
- To explore the I/O organizations in depth.
- To explore the memory organization.
- To understand the basic chip design and organization of 8086 with assembly language.

Course Outcomes: Students will be able to

1. After this course students understand in a better way the I/O and memory organization in depth.
2. Ability to understand the merits and pitfalls in computer performance measurements.
3. Identify the basic elements and functions of 8086 microprocessors.
4. Understand the instruction set of 8086 and use them to write assembly language programs.
5. Demonstrate fundamental understanding on the operation between the microprocessor and its interfacing devices.

UNIT-I

Basic Computer Organization: Functions of CPU, I/O Units, Memory: Instruction: Instruction FormatsOne address, two addresses, zero addresses and three addresses and comparison; addressing modes with numeric examples: Program Control- Status bit conditions, conditional branch instructions, Program Interrupts: Types of Interrupts.

UNIT-II

Input-Output Organizations: I/O Interface, I/O Bus and Interface modules: I/O Vs Memory Bus, Isolated Vs Memory-Mapped I/O, Asynchronous data Transfer- Strobe Control, Hand Shaking: Asynchronous Serial transfer- Asynchronous Communication interface, Modes of transfer Programmed I/O, Interrupt Initiated I/O, DMA; DMA Controller, DMA Transfer, IOP-CPU-IOP Communication, Intel 8089 IOP.

UNIT-III

Memory Organizations: Memory hierarchy, Main Memory, RAM, ROM Chips, Memory Address Map, Memory Connection to CPU, associate memory, Cache Memory, Data Cache, Instruction cache, Miss and Hit ratio, Access time, associative, set associative, mapping, waiting into cache, Introduction to virtual memory.

UNIT-IV

8086 CPU Pin Diagram: Special functions of general purpose registers, Segment register, concept of pipelining, 8086 Flag register, Addressing modes of 8086.

UNIT-V

8086-Instruction formats: assembly Language Programs involving branch & Call instructions, sorting, evaluation of arithmetic expressions.

Suggested Readings:

1. Computer system Architecture: Morris Mano (UNIT-1,2,3).

2. Advanced Micro Processor and Peripherals- Hall/ A K Ray(UNIT-4,5).

3. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.

4. Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition PHI/Pearson.

5. Fundamentals or Computer Organization and Design, - Sivaraama Dandamudi Springer Int. Edition.

6. Computer Architecture a quantitative approach, John L. Hennessy and David A. Patterson, Fourth Edition Elsevier.

7. Computer Architecture: Fundamentals and principles of Computer Design, Joseph D. Dumas II, BS Publication.

Course Code	Course Title						Core/Elective
PC234CM	**Operating System Concepts**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives:

- To introduce the concepts of OS structure and process synchronization.
- To study different memory management strategies.
- To familiarize the implementation of file system.
- To understand the principles of system security and protection.
- To discuss the design principles and structure of Windows 7 and Linux

Course Outcomes : Student will be able to

1. Evaluate different process scheduling algorithms.
2. Describe the steps in address translation and different page replacement strategies.
3. Compare different file allocation methods and decide appropriate allocation strategy for given type of file.
4. Explain the mechanisms available in an OS to control access to resource.

UNIT-I

Introduction to Operating Systems: OS structure and strategies, Process concepts, Multithreaded Programming, Process scheduling, Process synchronization, Deadlocks.

UNIT-II

Memory management strategies with example architectures: Swapping, Contiguous allocation, Paging, Segmentation, Segmentation with paging, Virtual memory management: Demand paging, Page replacement, Thrashing.

UNIT-III

File system interface: File concepts, Access methods and protection. File system implementation: File system structure, Allocation methods, Directory implementation of file systems, Mass storage structures, I/O systems.

UNIT-IV

System Protection: Principles and Domain, Access Matrix and implementation, Access control and access rights, Capability based systems, and Language based Protection System Security: Problem, Program threats, cryptography, user authentication, implementing security defenses, Firewalling, Computer Security Classification.

UNIT-V

Case Studies: The Linux System–Design principles, Kernel modules, Process management, Scheduling, Memory management, File systems, Input and Output, Inter process communication. Windows7–Design principles, System components, Terminal services and fast user switching File systems, Networking, Programmer interface.

Suggested Reading:

1. Abraham Silberschatz, Peter B Galvin, Operating System Concepts,9th Edition, Wiley, 2016
2. William Stallings, Operating Systems-Internals and Design Principles, 8thedition, Pearson, 2014
3. Andrew S Tanenbaum, Modern Operating Systems, 4th edition, Pearson, 2016.

Course Code	Course Title						Core/Elective
PC233CM	**Database Management Systems**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

- To Learn mathematical concepts as applied in computer
- To introduce three scheme architecture and DBMS functional components.
- To learn formal and commercial query languages of RDBMS
- To Study different file organization and indexing techniques
- To familiarize theory of serializablity and implementation of concurrency control, and recovery

Course Outcomes

1. Understand the mathematical foundations on which RDBMS are built
2. Model a set of requirements using the Extended Entity Relationship Model (EER), transform an EER model into a relational model and refine the relational model using theory of normalization
3. Develop Database application using SQL and Embedded SQL
4. Use the knowledge of file organization and indexing to improve database application performance
5. Understand the working of concurrency control and recovery mechanisms in RDBMS

UNIT-I

Introduction: Database System Application, Purpose of Database Systems, View of Values, Nested Sub-queries, Complex Queries views, Modification of the Database, Joined Relations

Data, Database Language, Relational Databases, Database Design, Object-Based and Semi-Structured Databases, Data Storages and Querying , Transaction Management ,Data Mining and Analysis, Database Architecture, Database Users and Administrators.

Database Design and the E-R Model: Overview of the Design Process, The Entity Relationship Model Constraints, Entity-Relationship Design issues, Weak Entity Sets Extended E-R Features Database Design for banking Enterprise, Reduction to Relational Schemas, Other Aspects of Database Design

UNIT-II

Relational Model: Structure of Relational Databases, Fundamental Relational-Algebra Operations, Additional Relational-Algebra Operations, Extended Relational-Algebra Operations, Null Values, Modification of the Databases

Structured Query Language: Data Definition, Basic Structure of SQL Queries, Set Operations, Aggregate Functions, Null

UNIT-III

Advanced SQL: SQL Data Types and Schemes, Integrity constraints, Authorization, Embedded SQL, Dynamic SQL, Functions and Procedural Constructs, Recursive Queries, Advanced SQL Features.

Relational Database Design: Features of Good Relational Design, Atomic Domains and First Normal Form, Functional Dependency Theory, Decomposition using Functional Dependencies.

UNIT-IV

Indexing and Hashing: Basic Concepts, Ordered Indices, B*-tree index files, B-tree index files, multiple key access, static hashing, dynamic hashing, comparison of ordered indexing and hashing bitmap indices.
Index definition in SQL transactions: Transaction concepts, transaction state, implementation of atomicity and durability, concurrent executions, serializability, recoverability, implementation of isolation, testing for serializability.

UNIT-V

Concurrency Control: Lock based protocols, timestamp based protocols, validation based protocols, multiple granularity, multi version schemes, deadlock handling, insert and delete operations, weak levels of consistency, concurrency of index structures.
Recovery system: Failure classification, storage structure, recovery and atomicity, log-based recovery, recovery with concurrent transactions, buffer management, failure with loss of non-volatile storage, advanced recovery techniques, remote backup systems.

Suggested Readings:

1. Abraham Silberschatz, Henry F Korth, S Sudarshan, Database System Concepts, McGraw-Hill, 6th Edition, 2010
2. Ramakrishnan, Gehrke, Database Management Systems, McGraw-Hill, 3rd Edition, 2003
3. Elmasri, Navathe, Somayajulu, Fundamentals of Database Systems, Pearson Education, 4th Edition, 2004.

Course Code	Course Title						Core/Elective
PC261CM	**Computer Organization & Microprocessor Lab**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**-**	**-**	**-**	**2**	**25**	**50**	**1**

Course Objectives

The objectives of the course are to impart knowledge of the:

- To become familiar with the architecture and Instruction set of Intel 8086 microprocessor.
- To provide practical hands on experience with Assembly Language Programming.
- To familiarize the students with interfacing of various peripheral devices with 8085 microprocessors.

Course Outcomes

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

1. Interpret the principles of Assembly Language Programming, instruction set in developing microprocessor based applications.
2. Develop Applications such as: 8-bit Addition, Multiplication, Division, array operations, swapping, negative and positive numbers.
3. Analyse the interfaces like serial ports, digital-to-analog Converters and analog-to-digital converters etc.
4. Build interfaces of Input-output and other units like stepper motor with 8086.
5. Analyse the function of traffic light controller.

List of Experiments:

1. Tutorials with 8086 kit / MASM software tool.

2. Fixed-point multiplication and division.

3. Floating-point multiplication and division.

4. Sorting hexadecimal array.

5. Code conversion from hexadecimal to decimal.

6. Sum of set of BCD numbers.

7. Searching.

8. Display a string of characters using 8279.

9.Interfacing traffic light controller using 8255.

10. Interfacing seven-segment LED using 8255.

11. Interfacing stepper motor using 8255.

12. Interfacing 8253 counter.

13. D/A conversion using 8255.

14. A/D conversion using 8255.

Suggested Readings:

1. Yu-cheng Liu, Glenn A. Gibson, "Microcomputer Systems: The 8086/8088 Family", 2nd Edition, PHI Learning 2011.

2. Douglas Hall. "Microprocessor and Interfacing programming and Hardware", Tata Mc Graw Hill, Revised 2nd Edition, 2007.

3. Brey B. Brey, "The Intel Microprocessor, 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium and Pentium Pro Processors-Architecture, Programming and interfacing", 4th Edition, Prentice Hall, 1993.

Course Code	Course Title						Core/Elective
PC264CM	**Operating Systems Lab**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**-**	**-**	**-**	**2**	**25**	**50**	**1**

Course Objectives

- To learn shell programming and the use of filters in the LINUX environment.
- To practice multithreaded programming.
- To implement CPU Scheduling Algorithms and memory management algorithms.

Course Outcomes

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

1. Write shell scripts for simple system administration tasks.
2. Write concurrent programs with synchronization constricts.
3. Compare the performance of various CPU Scheduling Algorithm.
4. Critically analyze the performance of the various Memory management algorithms

List of Experiments:

1-3. Memory Management Algorithms
4-5. Examples of Multithreading
6. Producer & Consumer problem using Semaphores and shared memory
7-8. Processor Scheduling algorithms
9. Dining Philosophers problem using Semaphores
10. Readers and Writers problem using Semaphores
11. Shell-programming exercises.

Course Code	Course Title						Core/Elective
PC263CM	**Database Management Systems Lab**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**-**	**-**	**-**	**2**	**25**	**50**	**1**

Course Objectives

- To practice various DDL commands in SQL
- To write simple and complex queries in SQL
- To familiarize PL/SQL

Course Outcomes

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

1. Design and implement a database schema for a given problem
2. Populate and query a database using SQL and PL/SQL
3. Develop multi-user database application using locks

Creation of database (exercising the commands for creation)

1. Simple to complex condition query creation using SQL Plus.
2. Usage of triggers and stored procedures
3. Creation of forms for student information, library information, pay roll etc.
4. Writing PL/SQL procedures for data validation.
5. Report generation using SQL reports.
6. Creating password and security features for applications.
7. Using of file locking, table locking facilities in applications.
8. Creation of small full-fledged database application spreading over 3 sessions.

Note: The creation of sample database for the purpose of the experiments is expected to be pre-decided by the instructor.

Course Code	Course Title						Core/Elective
PC262CM	**OOP using JAVA Lab**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**-**	**-**	**-**	**2**	**25**	**50**	**1**

Course Objectives

- Ability to learn the concept of classes, inheritance and abstract classes.
- To Learn to demonstrate multithreaded programs with synchronization.
- To Demonstrate real world applications using java collection frame work and I/O classes.
- To Model Event driven GUI programs using AWT/Swing.
- To build software development skills using java programming for real world applications.
- To implement frontend and backend of an application**.**
- To implement classical problems using java programming.

Course Outcomes

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

1. Able to understand the OOPS features.
2. Implement the concepts of Exception Handling in java Applications.
3. Read and write data using different Java I/O streams.
4. Create graphical user interfaces and Applets by applying the knowledge of Event Handling.
5. Create robust applications using Java standard class libraries and retrieve data from a database with JDBC.
6. Ability to solve real-world problems by designing user friendly GUI with befitting backend through the APIs of Java.

List of Experiments:

1) Write a Java program to illustrate the concept of class with method overloading.
2) Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java. util)
3) Write a Java program to illustrate the concept of Single level and Multi level Inheritance.
4) Write a Java program to demonstrate the Interfaces & Abstract Classes.
5) Write a Java program to implement the concept of exception handling.
6) Write a Java program to illustrate the concept of threading using Thread Class and runnable Interface.
7) Write a Java program to illustrate the concept of Thread synchronization.
8) Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
9) Write a Java program to illustrate collection classes like Array List, LinkedList, Tree map and Hash map.
10) Write a Java program to illustrate Legacy classes like Vector, Hashtable, Dictionary & Enumeration interface
11) Write a Java program to implement iteration over Collection using Iterator interface and List Iterator interface
12) Write a Java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
13) Write a Java program to illustrate the concept of I/O Streams

14) Write a Java program to implement serialization concept
15) Write a Java applet program to implement Colour and Graphics class
16) Write a Java applet program for handling mouse & key events
17) Write a Java applet program to implement Adapter classes
18) Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result.
19) Write an example for JDBC prepared statement with Result Set.
20) Program to get primary key value (auto-generated keys) from inserted queries using JDBC
21) Program to create a simple JList
22) java Program to create a simple checkbox using JCheckBox
23) Program to create a checkbox and ItemListener to it.
24) 1. Write Servlet application to print current date & time
    2. Html & Servlet Communication
    3. Auto refresh a page
    4. Demonstrate session tracking
    5. Select record from database
    6. Application for login page
    7. Insert record into database
    8. Count the visits on web page
    9. Insert teacher record in Database

SCHEME OF INSTRUCTION
BE (COMPUTER SCIENCE AND ENGINEERING)
AICTE MODEL CURRICULUM
Proposed from the Academic year 2018-19

CSE: SEMESTER – I (2018-19)

S. No.	Course Code	Course Title	Scheme of Instructions				Scheme of Examination			Credits
			L	T	Pr/ Drg	Contact Hours/ Week	CIE	SEE	Duration in Hours	
Theory Course										
1	**Three Week Induction Programme**						-	-	-	-
2	BS101MT	Mathematics-I	3	1	-	4	30	70	3	4
3	BS103PH	Physics	3	1	-	4	30	70	3	4
4	ES101EE	Basic Electrical Engineering	3	1	-	4	30	70	3	4
Practical/ Laboratory Course										
5	BS153PH	Physics Lab	-	-	3	3	25	50	3	1.5
6	ES151EE	Basic Electrical Engineering Lab	-	-	2	2	25	50	3	1
7	ES153CE	Engineering Graphics & Design	1	-	4	5	50	50	3	3
Total			**10**	**03**	**09**	**22**	**190**	**360**		**17.5**

CSE: SEMESTER – II

S. No	Course Code	Course Title	Scheme of Instruction				Scheme of Examination		
			L	T	P	Contact Hrs/Wk	CIE	SEE	Credits
1.	HS101EG	English	2	-	-	2	30	70	2
2.	BS102MT	Mathematics-II	3	1	-	4	30	70	4
3.	BS104CH	Chemistry	3	1	-	4	30	70	4
4.	ES102CS	Programming for Problem Solving	3	-	-	3	30	70	3
Practicals									
5.	HS151EG	English Lab			2		25	50	1
6.	BS154CH	Chemistry Lab			3	3	25	50	1.5
7	ES152CS	Programming for Problem Solving			4	4	25	50	2
8.	ES154ME	Workshop/ Manufacturing Process	1	-	4	5	50	50	3
Total			**12**	2	**13**	**25**	**245**	**480**	**20.5**

CSE: SEMESTER – III (2019-2020)

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	P/D	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	MC112CE	Environmental Science	2	-	-	2	30	70	3	-
2	MC113PY	Essence of Indian Traditional Knowledge	2	-	-	2	30	70	3	-
3	HS204ME	Operations Research	3	-	-	3	30	70	3	3
4	BS206BZ	Biology for Engineers	3	-	-	3	30	70	3	3
5	ES214EC	Basic Electronics	3	-	-	3	30	70	3	3
6	ES216EC	Digital Electronics	3	-	-	3	30	70	3	3
7	PC221CS	Data Structures and Algorithms	3	-	-	3	30	70	3	3
8	PC222CS	Discrete Mathematics	3	-	-	3	30	70	3	3
9	PC223CS	Programming Languages	3	-	-	3	30	70	3	3
Practical/ Laboratory Courses										
10	ES251EC	Basic Electronics Lab	-	-	2	2	25	50	3	1
11	PC252CS	Data Structures and Algorithms Lab	-	-	2	2	25	50	3	1
12	PC253CS	Advanced Computer Skills Lab	-	-	2	2	25	50	3	1
			25	**-**	**06**	**31**	**345**	**780**		**24**

CSE: SEMESTER – IV (2019-20)

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	P/D	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	MC111PO	Indian Constitution	2	-	-	2	30	70	3	-
2	HS201EG	Effective Technical Communication in English	3	-	-	3	30	70	3	3
3	HS202CM	Finance and Accounting	3	-	-	3	30	70	3	3
4	BS207MT	Mathematics – III (Probability & Statistics)	3	-	-	3	30	70	3	3
5	ES215EC	Signals and Systems	3	-	-	3	30	70	3	3
6	PC231CS	OOP using JAVA	3	-	-	3	30	70	3	3
7	PC232CS	Computer Organization	3	-	-	3	30	70	3	3
8	PC233CS	Database Management Systems	3	-	-	3	30	70	3	3
Practical/ Laboratory Courses										
9	PC261CS	Computer Organization Lab	-	-	2	2	25	50	3	1
10	PC262CS	OOP using JAVA Lab	-	-	2	2	25	50	3	1
11	PC263CS	Database Management Systems Lab	-	-	2	2	25	50	3	1
			23	**-**	**06**	**29**	**315**	**710**		**24**

CSE - SEMESTER - V (Proposed for the academic year 2020-21)

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Course										
1.	PC 501 CS Core-7	Software Engineering	3	1	-	4	30	70	3	3
2.	PC 502 CS Core-8	Operating Systems	3	1	-	4	30	70	3	3
3.	PC 503 CS Core-9	Automata Languages & Computation	3	1	-	4	30	70	3	3
4.	PE-I	Professional Elective-I	3	-	-	3	30	70	3	3
5.	PE-II	Professional Elective-II	3	-	-	3	30	70	3	3
6.	PE-III	Professional Elective-III	3	-	-	3	30	70	3	3
Practical/Laboratory Course										
7.	PC531 CS	Software Engineering Lab	-	-	2	2	25	50	3	1.5
8.	PC532 CS	Operating Systems Lab	-	-	2	2	25	50	3	1.5
9.	PW 533 CS	Mini Project	-	-	2	2	25	50	3	1
Total			**18**	**03**	**06**	**27**	**255**	**570**		**22**

Profession Elective – I	
Course Code	**Course Title**
PE 511 CS	Artificial Intelligence
PE 512 CS	Advanced Computer Architecture
PE 513 CS	Image Processing

Profession Elective – II	
Course Code	**Course Title**
PE 527 CS	Web Technologies
PE 528 CS	Embedded Systems
PE 529 CS	Graph Theory
PE 530 CS	Data Analytics

Profession Elective – III	
Course Code	**Course Title**
PE 523 CS	Block Chain Technologies
PE 524 CS	Information Retrieval Systems
PE 525 CS	Soft Computing
PE 526 CS	Computer Graphics

CSE - SEMESTER - VI (Proposed for the academic year 2020-21)

S. No	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	Contact Hrs/Wk	CIE	SEE	Duration in Hrs/Wk	
Theory Course										
1.	PC 601 CS Core-10	Compiler Design	3	1	-	4	30	70	3	3
2.	PC 602 CS Core-11	Computer Networks	3	1	-	4	30	70	3	3
3.	PC 603 CS Core 12	Design and Analysis of Algorithms	3	1	-	3	30	70	3	3
4.	PE –IV	Professional Elective -IV	3	-	-	3	30	70	3	3
5	PE –V	Professional Elective -V	3	-	-	3	30	70	3	3
6	OE-I	Open Elective-I	3	-	-	-	30	70	3	3
7	MC	Gender Sensitization	3	-	-	3	30	70	3	-
Practical/Laboratory Course										
8	PC631 CS	Compiler Design Lab	-	-	2	2	25	50	3	1
9	PC632 CS	Computer Networks Lab	-	-	2	2	25	50	3	1
10	PC 633 CS	Design and Analysis of Algorithms Lab	-	-	2	2	25	50	3	1
11	SI 671 IT	Summer Internship*	-	-	-	-	-	-	-	-
		Total	**18**	**3**	**6**	**27**	**205**	**570**		**21**

Profession Elective – IV	
Course Code	**Course Title**
PE 627CS	Advanced Operating Systems
PE 628 CS	Cloud Computing
PE 629 CS	Natural Language Processing
PE 630 CS	Machine Learning

Profession Elective – V	
Course Code	**Course Title**
PE 723 CS	Data Mining
PE 724 CS	Human Computer Interaction
PE 725 CS	Digital Forensics
PE 726 CS	Internet of things

Open Elective - I	
Course Code	**Course Title**
OE 601	Soft Skills & Interpersonal Skills
OE 602	Human Resource Development and Organizational Behaviour
OE 601	Cyber Law and Ethics

CSE - SEMESTER - VII

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	P/D	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	PC 701 CS Core-13	Information Security	3	-	-	4	30	70	3	3
2	PC 702 CS Core-14	CORE-14	3	1	-	4	30	70	3	3
3	PC 703 CS Core-15	Distributed Systems	3	1	-	3	30	70	3	3
4	PE-VI	Professional Elective – VI	3	-	-	3	30	70	3	3
5.	MC	NSS/Sports/Yoga	-	-	3	3	50	-	-	0
Practical/ Laboratory Courses										
5	PC 751 CS	Distributed Systems Lab	-	-	3	3	25	50	3	1
6	PC 752 CS	Data Mining Lab	-	-	3	3	25	50	3	1
7	PW 761 CS	Project Work – I	-	-	4	4	50	-	-	2
8	SI 762 CS	Summer Internship	`	-	-	-	50	-	-	2
			12	**02**	**13**	**28**	**290**	**310**		**18**

Profession Elective – VI	
Course Code	**Course Title**
PE 727 CS	Mobile Computing
PE 728 CS	Sematic Web
PE 729 CS	Web Security /Cyber Security

CS - SEMESTER - VIII

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	P/D	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	**OE-II**	**Open Elective – II**	3	-	-	3	30	70	3	3
2	**OE-III**	**Open Elective – III**	3	-	-	3	30	70	3	3
Practical/ Laboratory Courses										
3	**PW861 CS**	Project Work – II	-	-	16	16	50	100	-	8
			06	**-**	**16**	**22**	**160**	**240**	**06**	**14**

Open Elective – II			**Open Elective – III**		
S. No.	Course Code	Course Title	S. No.	Course Code	Course Title
1	**OE 771 CE**	Green Building Technologies	1	**OE 781 CE**	Road Safety Engineering
2	**OE 772 CS****	Data Science Using R Programming	2	**OE 782 IT****	Software Engineering
3	**OE 773 EC**	Fundamentals of IoT	3	**OE 783 EC**	Principles of Electronic Communications
4	**OE 774 EE**	Non-Conventional Energy Sources	4	**OE 784 EE**	Illumination and Electric Traction systems
5	**OE 775 ME**	Entrepreneurship	5	**OE 785 ME**	Mechatronics
6	**OE 876 IT**	Cyber Security			

S.No	**Course Work-Subject Area**	**Credits/Semester**								**Total Credits**
		I	**II**	**III**	**IV**	**V**	**VI**	**VII**	**VIII**	
1.	Humanities and Social Sciences (HS) - **AICTE -12**	**-**	**3** (1+1)	**6** **(3+3)**	**3** **(1)**		**-**	**-**	**-**	**11**
2.	Basic Sciences (BS) **AICTE -24**	9.5 (2+1)	9.5 (2+1)	**3** (1)	**3** (1)	**-**	**-**	**-**	**-**	**25**
3.	Engineering Sciences (ES) **AICTE-29**	**8** (2+1)	**8** (1+2)	**7** (2+1)	3 (1)	**-**	**-**	**-**	**-**	**26**
4.	Professional Subjects- Core (PC) **AICTE-49**	**-**	**-**	**11** (3+2)	**12** (3+3)	**12** (3+3)	**12** (3+3)	**11** (3+2)		**58**
5.	Professional Subject- Electives (PE*) **AICTE-18**	**-**	**-**	**-**	-	9 (3)	**6** (2)	**3** (1)		**18**
6.	Open Subjects- Electives (OE) **AICTE-12**	**-**	**-**	**-**	-	**-**	**3** (1)	-	**6** (2)	**09**
7.	Mini Project, Project Work-I and II and Internship **AICTE-15**	**-**	**-**	**-**	-	1	**-**	**4**	**8**	**13**
	TOTAL	**17.5**	**20.5**	**24**	**24**	**21**	**21**	**18**	**14**	**160**
8.	Mandatory Courses (MC) (*Non-Credit*) **8 Units**			**3** (1)	**6** **(2)**	3 (3U)				**12**

SCHEME OF INSTRUCTION
BE (COMPUTER SCIENCE AND ENGINEERING)
AICTE MODEL CURRICULUM
CSE - SEMESTER - V (Proposed for the academic year 2020-21)

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	Contact Hrs/W	CIE	SEE	Duration in Hrs	
Theory Course										
1.	PC 501 CS Core-7	Software Engineering	3	1	-	4	30	70	3	3
2.	PC 502 CS Core-8	Operating Systems	3	1	-	4	30	70	3	3
3.	PC 503 CS Core-9	Automata Languages & Computation	3	1	-	4	30	70	3	3
4.	PE-I	Professional Elective-I	3	-	-	3	30	70	3	3
5.	PE-II	Professional Elective-II	3	-	-	3	30	70	3	3
6.	PE-III	Professional Elective-III	3	-	-	3	30	70	3	3
Practical/Laboratory Course										
7.	PC531 CS	Software Engineering Lab	-	-	2	2	25	50	3	1.5
8.	PC532 CS	Operating Systems Lab	-	-	2	2	25	50	3	1.5
9.	PW533 CS	Mini Project	-	-	2	2	25	50	3	1
Total			**18**	**03**	**06**	**27**	**255**	**570**		**22**

Profession Elective – I	
Course Code	**Course Title**
PE 511 CS	Artificial Intelligence
PE 512 CS	Advanced Computer Architecture
PE 513 CS	Image Processing

Profession Elective – II	
Course Code	**Course Title**
PE 527 CS	Web and Internet Technologies
PE 528 CS	Embedded Systems
PE 529 CS	Graph Theory
PE 530 CS	Data Analytics

Profession Elective – III	
Course Code	**Course Title**
PE 523 CS	Block Chain Technologies
PE 524 CS	Information Retrieval Systems
PE 525 CS	Soft Computing
PE 526 CS	Computer Graphics

CSE - SEMESTER - VI (Proposed for the academic year 2020-21)

S. No	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	Contact Hrs/Wk	CIE	SEE	Duration in Hrs/Wk	
Theory Course										
1.	PC 601 CS Core-10	Compiler Design	3	1	-	4	30	70	3	3
2.	PC 602 CS Core-11	Computer Networks	3	1	-	4	30	70	3	3
3.	PC 603 CS Core 12	Design and Analysis of Algorithms	3	1	-	3	30	70	3	3
4.	PE –IV	Professional Elective - IV	3	-	-	3	30	70	3	3
5	PE –V	Professional Elective -V	3	-	-	3	30	70	3	3
6	OE-I	Open Elective-I	3	-	-	-	30	70	3	3
Practical/Laboratory Course										
7	PC631 CS	Compiler Design Lab	-	-	2	2	25	50	3	1
8	PC632 CS	Computer Networks Lab	-	-	2	2	25	50	3	1
9	PC 633 CS	Design and Analysis of Algorithms Lab	-	-	2	2	25	50	3	1
10	SI 671 IT	Summer Internship*	-	-	-	-	-	-	-	-
Total			**18**	**3**	**6**	**27**	**205**	**570**		**21**

Profession Elective – IV

Course Code	Course Title
PE 627CS	Advanced Operating Systems
PE 628 CS	Cloud Computing
PE 629 CS	Speech and Natural Language Processing
PE 630 CS	Machine Learning

Profession Elective – V

Course Code	Course Title
PE 651 CS	Data Mining
PE 652CS	Human Computer Interaction
PE 653 CS	Digital Forensics
PE 654 CS	Internet of Things

Open Elective - I	
Course Code	**Course Title**
OE 601	Soft Skills & Interpersonal Skills
OE 602	Human Resource Development and Organizational Behaviour
OE 601	Cyber Law and Ethics

Course Code	Course Title						Core/ Elective
PC 501 CS	**SOFTWARE ENGINEERING**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	1	-	-	30	70	3

Course Objectives

- To introduce the basic concepts of software development processes from defining a product to shipping and maintaining
- To impart knowledge on various phases, methodologies and practices of software development
- To understand the importance of testing in software development, study various testing strategies along with its relationship with software quality and metrics

Course Outcomes

Student will be able to

- Acquired working knowledge of alternative approaches and techniques for each phase of software development
- Judge an appropriate process model(s) assessing software project attributes and analyze necessary requirements for project development eventually composing SRS
- Creation of visual models to describe (non-) algorithmic solutions for projects using various design principles.
- Acquire skills necessary as an independent or as part of a team for architecting a complete software project by identifying solutions for recurring problems exerting knowledge on patterns.
- Concede product quality through testing techniques employing appropriate metrics by understanding the practical challenges associated with the development of a significant software system.

UNIT-I

Introduction to Software Engineering:A generic view of Process: Software Engineering, Process Framework, CMM Process Patterns, Process Assessment.

Process Models: Prescriptive Models, Waterfall Model, Incremental Process Models, Evolutionary Process Models, Specialized Process Models, The Unified Models, Personal and Team Process Models, Process Technology, Product and Process.

An Agile view of Process: Introduction to Agility and Agile Process, Agile Process Models.

UNIT-II

Software Engineering Principles: SE Principles, Communication Principles, Planning Principles, Modeling Principles, Construction Principles, Deployment.

System Engineering: Computer-based Systems, The System Engineering Hierarchy, Business Process Engineering, Product Engineering, System Modeling.

Requirements Engineering: A Bridge to Design and Construction, Requirements Engineering Tasks, Initiating Requirements Engineering Process, Eliciting Requirements, Developing Use-Cases, Building the Analysis Model, Negotiating Requirements, Validating Requirements.

UNIT-III

Building the Analysis Model: Requirements Analysis Modeling Approaches, Data Modeling Concepts, Object-Oriented Analysis, Scenario-based Modeling, Flow-oriented Modeling, Class-based Modeling, Creating a Behavioral Model.

Design Engineering: Design within the context of SE, Design Process and Design Quality, Design Concepts, The Design Model, Pattern-based Software Design.

UNIT-IV

Creating an Architectural Design: Software Architecture, Data Design, Architectural Styles and Patterns, Architectural Design.

Modeling Component-Level Design: Definition of Component, Designing Class-based Components, Conducting Component-level Design, Object Constraint Language, Designing Conventional Components.

Performing User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, Design Evaluation.

UNIT-V

Testing: Strategies: A Strategic Approach to Conventional Software Testing, Test Strategies for O-O Software. **Tactics:** Software Testing Fundamentals, Black-box and White-box Testing, Basis Path Testing, Control Structure Testing, O-O Testing Methods.

Debugging: Debugging Techniques, The Art of Debugging.

Product Metrics: A Framework for Product Metrics, Metrics for each phase of software development.

Software Quality: Definition, **Quality Assurance:** Basic Elements, Formal Approaches, Statistical Software Quality Assurance, Software Reliability, ISO9000 Quality Standards, SQA Plan.

Suggested Books:

1. Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th Edition, McGraw Hill, 2009
2. Ali Behforooz and Frederick J. Hudson, *Software Engineering Fundamentals*, Oxford University Press, 1996
3. PankajJalote, *An Integrated Approach to Software Engineering*, 3rd Edition, Narosa Publishing House, 2008

Course Code	Course Title						Core/ Elective
PC 502 CS	**OPERATING SYSTEMS**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	1	-	-	30	70	3

Course Objectives

- To learn the fundamentals of Operating Systems.
- To learn the mechanisms of OS to handle processes and threads and their communication
- To learn the mechanisms involved in memory management in contemporary OS
- To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection
- To know the components and management aspects of concurrency management

Course Outcomes

- Identify System calls and evaluate process scheduling criteria of OS.
- Develop procedures for process synchronization of an OS.
- Demonstrate the concepts of memory management and of disk management
- Solve issues related to file system interface and implementation, I/O systems
- Describe System model for deadlock, Methods for handling deadlocks.

UNIT-I

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine.

UNIT-II

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching

Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads,

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling Criteria, Scheduling algorithms, multiprocessor scheduling

UNIT-III

Process Synchronization: Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Peterson’s Solution, classical problems of synchronization: The Bounded buffer problem, Producer\Consumer Problem, reader’s & writer problem, Dinning philosopher’s problem. Semaphores, Event Counters, Monitors, Message Passing,

Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Methods for Handling: Deadlocks: Deadlock prevention, Deadlock Avoidance: Banker’s algorithm, Deadlock detection and Recovery.

UNIT-IV

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation, fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, structure of page table, Protection and sharing, Disadvantages of paging.

Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms, Trashing

UNIT-V

I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software,

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods, Free-space management, directory implementation, efficiency and performance.

Secondary-Storage Structure: Disk structure, Disk scheduling algorithms, Disk Management, RAID structure

Suggested books:

1. AviSilberschatz, Peter Galvin, Greg Gagne, *Operating System Concepts Essentials*, 9th Edition, Wiley Asia Student Edition, 2017.
2. William Stallings, *Operating Systems: Internals and Design Principles*, 5th Edition, Prentice Hall of India, 2016.
3. Maurice Bach, *Design of the Unix Operating Systems*, 8th Edition,Prentice-Hall of India, 2009.
4. Daniel P. Bovet, Marco Cesati, *Understanding the Linux Kernel*, 3rd Edition, , O'Reilly and Associates.

Course Code	Course Title						Core/ Elective
PC 503 CS	**AUTOMATA LANGUAGES & COMPUTATION**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	1	-	-	30	70	3

Course Objectives

- Develop a formal notation for strings, languages and machines.
- Design finite automata to accept a set of strings of a language.
- Prove that a given language is regular and apply the closure properties of languages.
- Design context free grammars to generate strings from a context free language and Convert them into normal forms.
- Prove equivalence of languages accepted by Push down Automata and languages generated by context free grammars
- Identify the hierarchy of formal languages, grammars and machines.
- Distinguish between computability and non-computability and Decidability and undecidability.

Course Outcomes

- Write a formal notation for strings, languages and machines.
- Design finite automata to accept a set of strings of a language.
- For a given language determine whether the given language is regular or not.
- Design context free grammars to generate strings of context free languages.
- Determine equivalence of languages accepted by Pushdown Automata and languages generated by context free grammars
- Write the hierarchy of formal languages, grammars and machines.
- Distinguish between computability and non-computability and Decidability and undecidability.

UNIT-I

Introduction: Finite state automata, Non-deterministic finite state automata, FA with €-transitions, Regular expressions, Applications of FA, Properties of regular sets, Pumping Lemma, Closure properties, Myhill-Nerode Theorem, Minimization of FA.

UNIT-II

Context Free Grammars and Languages: Derivations, Parse-trees, Ambiguity in Grammars and Languages. Pushdown Automata–Definitions, The languages of PDA, Equivalence of PDAs and CFGs, Deterministic Pushdown Automata**.**

UNIT-III

Properties of CFLs: Normal forms for CFGs, Pumping Lemma, Closure properties,

Deterministic Context Free Languages, Decision properties.

UNIT-IV

Turing Machines: Introduction, Computational Languages and Functions, Techniques for construction of Turing machines. Modifications of TM, TM as enumerator, Restricted TM.

UNIT-V

Undecidability: Recursive and Recursively enumerable languages, UTM and undecidable problem, Rice Theorem, Post's correspondence problem. Chomsky's Hierarchy–Regular grammars, Unrestricted grammar, CSL, Relationship between classes of languages.

Suggested Books:

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd Edition, Pearson Education Asia,2007
2. John Martin, *Introduction to Languages and The Theory of Computation*, 3rd Edition,Tata McGraw Hill,2013.

Course Code	Course Title						Core/ Elective
PC 531 CS	**SOFTWARE ENGINEERING LAB**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	25	50	2

Course Objectives

- To understand the software engineering methodologies for project development.
- To gain knowledge about open source tools for Computer Aided Software Engineering (CASE).
- To develop test plans and test cases to perform various testing.

Course Outcomes

Student will be able to:

- Analyze and design software requirements in an efficient manner.
- Use open source case tools to develop software
- Implement the design , debug and test the code

I. FORWARD ENGINEERING

Students have to form a team with a batch size of two or three and take up a **case study based project** to analyze, plan, design UML models and create a prototypical model (identifying deliverables) by coding the developed designs and finally documenting considering any one example of the following domains:-

1. Academics (Course Registration System, Student marks analyzing system)
2. Health Care (Expert system to prescribe medicines for given symptoms, Remote Diagnostics, Patient/Hospital Management System)
3. Finance (Banking:ATM/NetBanking, UPI:PayTM/PhonePay, Stocks:Zerodha)
4. E-Commerce (various online shopping portals like FlipKart/Amazon/Myntra)
5. Logistics (Postal/Courier:IndiaPost/DTDC/UPS/FedEx, Freight:Maersk)
6. Hospitality (Tourism Management:Telangana Tourism/Incredible India, Event Management: MeraEvents/BookMyShow/Explara/EventBrite)
7. Social Networking (LinkedIn, FaceBook, Shaadi.com, BharatMatrimony, Tinder)
8. Customer Support (Banking Ombudsman,Indian Consumer Complaints Forum)
9. Booking/Ticketing(Food:Zomato/Swiggy/BigBasket/Grofers/JioMart, Hotel:OYO/Trivago or Travel: {Cars:Uber/OLA/Zoom, Railways:IRCTC, Buses:OnlineTSRTC/RedBus/AbhiBus, Flights:MakeMyTrip/Goibibo, Ships:Lakport})

II. REVERSE ENGINEERING: Students have to refer any project repository:GitLab/GitHub, execute the code in order to observe its functionalities/features/requirements and by the help of any tool derive the designs from the code for understanding the relationships among various subsystems/classes/components and if the tool partially generates models then identify by associating elements to judge/mark the appropriate relationships.

III. TESTING: Prepare Test Plan and develop Test Case Hierarchy to monitor or uncover/report errors using manual/automated testing tools

Software Required: *StarUML/Umbrello, NetBeans/Eclipse IDE, XAMPP/MEAN stack, JUnit, JMeter, Selenium, Bugzilla*

Course Code	Course Title						Core/ Elective
PC 532 CS	**OPERATING SYSTEMS LAB**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	25	50	2

Course Objectives

- Learn different types of CPU scheduling algorithms
- Demonstrate the usage of semaphores for solving synchronization problem
- Understand memory management techniques and different types of fragmentation that occur in them and various page replacement policies
- Understand Banker's algorithm used for deadlock avoidance
- Learn various disk scheduling algorithms.

Course Outcomes

- Evaluate the performance of different types of CPU scheduling algorithms
- Implement producer-consumer problem, reader-writers problem, Dining philosopher's problem
- Simulate Banker's algorithm for deadlock avoidance
- Implement paging replacement and disk scheduling techniques
- Use different system calls for writing application programs.

I. CASE STUDY

Perform a case study by installing and exploring various types of operating systems on a physical or logical (virtual) machine

II. List of Experiments (preferred programming language is C)

1. Write a C programs to implement UNIX system calls and file management
2. Write C programs to demonstrate various process related concepts.
3. Write C programs to demonstrate various thread related concepts.
4. Write C programs to simulate CPU scheduling algorithms: FCFS, SJF, Round Robin
5. Write C programs to simulate Intra & Inter-Process Communication (IPC) techniques: Pipes, Messages Queues, Shared Memory.
6. Write C programs to simulate solutions to Classical Process Synchronization Problems: Dining Philosophers, Producer-Consumer, Readers-Writers
7. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance.
8. Write C programs to simulate Page Replacement Algorithms: FIFO, LRU
9. Write C programs to simulate implementation of Disk Scheduling Algorithms: FCFS, SSTF

Course Code	Course Title						Core/ Elective
PW 533 CS	**MINI PROJECT**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	50	-	

Course Objectives: To prepare the students

- To enhance practical and professional skills.
- To familiarize tools and techniques of systematic literature survey and documentation
- To expose the students to industry practices and team work.
- To encourage students to work with innovative and entrepreneurial ideas

Course Outcomes

- Demonstrate the ability to synthesize and apply the knowledge and skills acquired in the academic program to the real-world problems.
- Evaluate different solutions based on economic and technical feasibility
- Effectively plan a project and confidently perform all aspects of project management
- Demonstrate effective coding, written, presentation and oral communication skills

The students are required to carry out mini projects in any of the areas such as Data Structures, Microprocessors and Interfacing, Database Management Systems, Operating Systems, Design and Analysis of Algorithms, Software Engineering, Data Communications, Web Programming & Services, Computer Networks, Compiler Construction, and Object Oriented System Development.

Problems Statements are suggested to be taken from Smart India Hackathon (SIH) Portal invited from the Ministries / PSUs / MNCs / NGOs to be worked out through.

The project could be classified as hardware, software, modeling, simulation etc. The project should involve one or many elements of techniques such as analysis, design, and synthesis.

The department will appoint a project coordinator who will coordinate the following:

1. Grouping of students (maximum of 3 students in a group)
2. Allotment of projects and project guides.
3. All projects allotment is to be completed by the 4th week of the semester so that the students get sufficient time for completion of the project.
4. Disseminate guidelines given by monitoring committee comprising of senior faculty members to the students and their guides.

Sessional marks are to be awarded by the monitoring committee.

Common norms will be established for the final presentation and documentation of the project report by the respective departments.

Students are required to submit a presentation and report on the mini project at the end of the semester.

Course Code	Course Title						Core/ Elective
PE 511 CS	**ARTIFICIAL INTELLIGENCE**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- ➢ Understand the importance of the field of AI by discussing its history and various applications.
- ➢ Learn about one of the basic applications of A.I, search state formulations.
- ➢ Learn methods of expressing knowledge by a machine with appropriate reasoning and different mathematics involved behind it
- ➢ Learn how to reason when an agent has only uncertain information about its task.
- ➢ Know various supervised and unsupervised learning algorithms

Course Outcomes

Upon completion of the course, the students will be able to:

- ➢ Formalize a problem in the language/framework of different AI methods
- ➢ Illustrate basic principles of AI in solutions that require problem solving, search, inference
- ➢ Represent natural language/English using Predicate Logic to build knowledge through various representation mechanisms
- ➢ Demonstrate understanding of steps involved in building of intelligent agents, expert systems, Bayesian networks
- ➢ Differentiate between learning paradigms to be applied for an application

UNIT-I

Problem Solving & Search: Introduction- What is intelligence? Foundations of artificial intelligence (AI). History of AI, Structure of Agents;

Problem Solving - Formulating problems, problem types, states and operators, state space;

Search Strategies. - Informed Search Strategies- Best first search, A* algorithm, heuristic functions, Iterative deepening A*;

Adversarial Search/ Game playing - Perfect decision game, imperfect decision game, evaluation function, alpha-beta pruning;

UNIT-II

Knowledge, Reasoning & Planning : Reasoning - Knowledge based agent, Propositional Logic, Inference, Predicate logic (first order logic), Resolution

Structured Knowledge Representation – Frames, Semantic Nets

Planning - A Simple Planning Agent, Form Problem Solving to Planning, Basic representation of plans, partial order planning, hierarchical planning

UNIT-III

Expert Systems, Reasoning with Uncertainty: Expert System and Applications: Introduction, Phases in Building Expert Systems, Expert System Architecture, Applications;

Uncertainty - Basic probability, Bayes rule, Belief networks, Inference in BayesianNetworks**,** Fuzzy sets and fuzzy logic: Fuzzy logic system architecture, membership function;

Decision Making- Utility theory, utility functions;

UNIT-IV

Learning: Machine-Learning Paradigms: Introduction, Machine Learning Systems, Supervised and Unsupervised Learning, Inductive Learning, Learning Decision Trees

Artificial Neural Networks: Introduction, Artificial Neural Networks, Single-Layer Feed-Forward Networks, Multi-Layer Feed-Forward Networks

Reinforcement learning – Learning from rewards, Passive and Active reinforcement learning, Applications

UNIT-V

Communicating & Perceiving: Introduction to NLP- Progress & applications of NLP, Components of NLP, Grammars, Parsing

Automatic Speech Recognition (ASR) – Speech Processing, Ex: DRAGON, HARPY, **Machine Vision –** Applications, Basic Principles of Vision, Machine vision techniques: Low, Middle and High level vision

AI Today & Tomorrow - Achievements, ubiquitous AI

Suggested Readings:

1. Stuart Russell and Peter Norvig. *Artificial Intelligence – A Modern Approach*, 3rd Edition, Pearson Education Press, 2009.
2. Kevin Knight, Elaine Rich, B. Nair, *Artificial Intelligence*, 3rd Edition, McGraw Hill, 2008.
3. Nils J. Nilsson, *The Quest for Artificial Intelligence*, Cambridge University Press, 2009

Course Code	Course Title						Core/ Elective
PE 512 CS	**ADVANCED COMPUTER ARCHITECTURE**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
--	3	-	-	-	30	70	-

Course Objectives

- An overview of computer architecture, which stresses the underlying design principles and the impact of these principles on computer performance. General topics include design methodology, processor design, control design, memory organization, system organization, and parallel processing.

Course Outcomes

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

- Know the classes of computers, and new trends and developments in computer architecture
- Understand pipelining, instruction set architectures, memory addressing.
- Understand the performance metrics of microprocessors, memory, networks, and disks
- Understand the performance and efficiency in advanced multiple-issue processors.
- Understand symmetric shared-memory architectures and their performance.

UNIT-I

Introduction - What is computer architecture? Software-hardware interface. Performance and Power. Performance metrics. Performance measurement. Benchmark programs.

UNIT-II

Instructions- Instruction Set. Operations. Operands and addressing modes. Role of compilers and system software. Understanding implementation of function calls and returns, array references, pointers.

UNIT-III

Computer Arithmetic- Signed integers. Floating point. Rounding and accuracy. Addition and Subtraction. Multiplication. Division

Processor - Data path elements. Data path control.

UNIT-IV

Pipelining - Speedup. Pipeline hazards. Stalling. Forwarding. Branch prediction. Exceptions. Speculation. Multiple issue.

Dynamic scheduling; Cache memory- Locality of reference. Cache organization and access. Multilevel caches. Performance. Cache coherence.

UNIT-V

Virtual Memory- Hardware support for address translation, page fault handling. Translation look aside buffer, Hardware-software interface.

Input/Output- Hard disk. Flash memory. I/O interfacing. Memory mapped I/O. Interrupt driven I/O. Direct memory access. Redundant arrays of inexpensive disks; Introduction to Multi-core architecture, Multi-processors. Clusters.

Suggested Readings:

1. David A. Patterson and John L. Hennessy, *Computer Organization and Design: The Hardware and Software Interface,* Morgan Kaufmann Publishers, 4th Edition.(2009)
2. John L. Hennessy and David A. Patterson, Computer Architecture: *A Quantitative Approach*, Morgan Kaufmann Publishers (2007)

Course Code	Course Title						Core/ Elective
PE 513 CS	**IMAGE PROCESSING**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
DS,DM	3	-	-	-	30	70	-

Course Objectives

Objectives of the course

- To introduce basics of visual perception, sampling, quantization and representation of digital images
- To introduce spatial domain and frequency domain filtering techniques necessary for image processing operations.
- To learn advanced image analysis techniques such as image restoration, image compression, image segmentation
- To learn techniques of multi resolution methods, wavelets and morphological processing.
- To understand the applications of image processing.

Course Outcomes

- Understand the basic image enhancement techniques in spatial & frequency domains.
- Understand the basics of multi-resolution techniques.
- Understand the basics of segmentation methods.
- Apply this concept for image handling in various fields.
- Knowledge about Morphological operations.

UNIT-I

Fundamentals of Image Processing: Introduction, examples, fundamental steps, components, elements of visual perception, light and electromagnetic spectrum, image sensing and acquisition, image sampling and quantization, basic relationships between pixels.
Intensity Transformations And Spatial Filtering: Background, some basic intensity transformation functions, histogram processing, fundamentals of spatial filtering, smoothing spatial filters, sharpening spatial filters, combining spatial enhancement methods.

UNIT-II

Filtering In The Frequency Domain: Background, preliminary concepts, sampling and Fourier transform of sampled functions, discrete Fourier transform (DFT) of one variable, extension to functions of two variables, some properties of the 2-D discrete Fourier transform, basics of filtering in the frequency domain, image smoothing, image sharpening, homo- morphic filtering.

UNIT –III

Image Restoration: Noise models, restoration in the presence of noise only-spatial filtering, periodic noise reduction by frequency domain filtering, linear degradation, position-invariant degradation, estimating the degradation function, inverse filtering, minimum mean square error filtering, constrained least squares filtering, geometric mean filter.

UNIT - IV

Wavelets And Multi Resolution Processing: Background, multi-resolution expansions, wavelet transforms in one dimension, the fast wavelet transform, wavelet transforms in two dimensions, wavelet packets.

Image Compression: Fundamentals, image compression models, elements of information theory, error free compression, lossy compression, image compression standards.

UNIT-V

Image Segmentation: Fundamentals, point, line and edge detection, thresholding, region-based segmentation, segmentation using morphological watersheds, the use of motion in segmentation.

Morphological Image Processing: Preliminaries, erosion and dilation, opening and closing, the Hit-or-Miss transformation, some basic morphological algorithms, some basic gray-scale morphological algorithms.

Suggested Readings:

1. Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, PHI Learning Pvt. Limited, 3rd Edition, 2008.
2. Rafael C.Gonzalez, Richard E.Woods and Steven L.Eddins, *Digital Image Processing Using MATLAB*, 2nd Edition, McGraw Hill, 2010.
3. AL. Bovik, *The Essential Guide to Image processing*, 2nd Edition, Elsevier, 2009.
4. Anil K.Jain, “Fundamentals of Digital Image Processing”, PHI, 2006.
5. William K. Pratt, *Digital Image Processing*, John Wiley & Sons, Inc., 3rd Edition, 2001

<table>
<tr><td>Course Code</td><td colspan="6">Course Title</td><td>Core/ Elective</td></tr>
<tr><td><b>PE 521 CS</b></td><td colspan="6"><b>WEB &amp; INTERNET TECHNOLOGY</b></td><td><b>ELECTIVE</b></td></tr>
<tr><td rowspan="2">Prerequisite</td><td colspan="4">Contact Hours Per Week</td><td rowspan="2">CIE</td><td rowspan="2">SEE</td><td rowspan="2">Credits</td></tr>
<tr><td>L</td><td>T</td><td>D</td><td>P</td></tr>
<tr><td>C, C++, Java, DC</td><td>3</td><td>-</td><td>-</td><td>-</td><td>30</td><td>70</td><td>-</td></tr>
</table>

Course Objectives

- Learn various client side technologies for developing web based applications.
- Learn the concepts of JavaScript and Angular JS for adding rich GUI.
- To Know about XML applications with DTD and Schema.
- To familiarize the concepts about Servlets and JSPs in dynamic web applications.
- To learn how to establish database connectivity in web applications.

Course Outcomes

- Understand the concepts of HTML and CSS.
- Acquire the knowledge to build AJAX based applications using Javascript.
- Understand and apply the concepts of servlet framework
- Implement JSP to build interactive web applications
- Acquire the knowledge of database connectivity in web applications

UNIT-I

A Brief Introduction to Internet, The World Wide Web, Web Browsers, Web Servers, Uniform Resource Locators, MIME, HTTP

HTML5: Evolution of HTML and XHTML, Basic Syntax, Document Structure, Links, Images, Multimedia, Lists, Tables, Creating Forms. Cascading Style sheets.

UNIT-II

JavaScript: Overview, Object Orientation and JavaScript, Syntactic Characteristics, Primitives, Operators, Expressions, Input and Output, Control Statements, Objects Creation and modification, Arrays, Functions, Constructors, Pattern Matching. Manipulating DOM, HTML DOM Events, Basics of AJAX with example.

UNIT-III

XML: Introduction to XML, Syntax, XML document structure, Document Type Definition, Name spaces, XML Schemas, Display in raw XML documents, Displaying XML documents with CSS, XPath Basics, XSLT, XML Processors.

J2EE: Exploring Enterprise architecture styles, Features of EE platform, Web servers and application servers.

Database programming with JDBC: JDBC Drivers, Exploring JDBC Processes with the java.sql Package.

UNIT-IV

Servlets Technology: Exploring the Features of Java Servlet, Exploring the Servlet API, Explaining the Servlet Life Cycle, Creating a Sample Servlet, Working with ServletConfig and ServletContext Objects, Implementing Servlet Collaboration, Exploring the Session Tracking Mechanisms.

UNIT-V

JSP Technology: Advantages of JSP over Java Servlet, Architecture of a JSP Page, Life Cycle of a JSP Page, Working with JSP Basic Tags and Implicit Objects, Working with Action Tags in JSP, Exploring EL, Exploring the Elements of Tag Extensions, Tag Extension API, Working with Simple Tag Handlers, Accessing Database from Servlet and JSP.

Suggested Readings :

1. Robert W.Sebesta: *Programming the World Wide Web,* 4th Edition, Pearson Education, 2009
2. Java Server Programming Java EE7 (J2EE 1.7): Black Book, (2014), Dreamtech Press
3. Porter Scobey, PawanLingras: *Web Programming and Internet Technologies an E-Commerce Approach*, 2nd Edition, Jones & Bartlett Learning, 2009.
4. Bryan Basham, Kathy Sierra, Bert Bates: Head first Servlets & JSP, 2nd edition, OREILLY, 2008.

Course Code	Course Title						Core/ Elective
PE 522 CS	**EMBEDDED SYSTEMS**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To provide an overview of Design Principles of Embedded System.
- Understand the fundamentals of Microcontroller based systems, basic hardware components, selection methods and attributes of an embedded system.
- To introduce and discuss Interfacing of various real world devices with 8051 microcontroller
- Comprehend the real time operating system used for the embedded system
- To expose students to the recent trends in embedded system design.

Course Outcomes

- Demonstrate the role of individual components involved in a typical embedded system.
- Describe the architectural features and instructions of Intel 8051 Microcontroller
- Apply the knowledge gained for Programming ARM for different applications.
- Expected to visualize the role of Real time Operating Systems in Embedded Systems
- Interpret the basic hardware components and their selection method based on the characteristics and attributes of an embedded system.

UNIT-I

Embedded Computing: Introduction, Complex Systems and Microprocessor; Embedded

System Design Process, Design Examples.

The 8051 Microcontrollers: Introduction, 8051 Micro Controller Hardware, Input/Output Ports and Circuits, External Memory, Counter and Timers, Serial data Input/Output, interrupts.

UNIT-II

Basic Assembly Language Programming Concepts: Assembly Language Programming

Process, Programming Tools and Techniques, Programming the 8051, Data Transfer

and Logical Instructions. Arithmetic Operations, Decimal Arithmetic, Jump and Call Instructions, Further Details on Interrupts.

UNIT-III

Interfacing real world devices with 8051 microcontroller:

Analog to Digital converters (ADC) & Digital to Analog Converter (DAC) basics. ADC, DAC and Temperature Sensor interfacing with 8051 microcontroller. LCD and Matrix Keyboard interfacing with 8051microcontroller.

UNIT-IV

Introduction to Real-Time Operating Systems: Tasks and Task States, Tasks and Data, Semaphores, Shared Data, Message Queues, Mailboxes and Pipes, Timer Functions, Events, Memory Management, Interrupt Routines in an RTOS Environment.

UNIT-V

Embedded Software Development Tools: Host and Target machines, Linker/Locators for Embedded Software, Getting Embedded Software into the Target System.

Debugging Techniques: Testing on Host Machine, Using Laboratory Tools, An Example System.

Suggested Readings:

1. Wayne Wolf , Computers as Components-Principles of Embedded Computer System Design, Morgan Kaufmann publishers, Third edition, 2012.
2. Mohamed Ali Mazidi, Janice GillispieMazidi, RolinMcKinlay, *The 8051 Microcontroller and Embedded Systems: Using Assembly and C*, 2nd Edition, Pearson education, 2011.
3. Raj Kamal, Embedded Systems: Architecture, Programming and Design, 3rd Edition, McGraw Hill Education (India), 2014.

Course Code	Course Title						Core/ Elective
PE 523 CS	**GRAPH THEORY**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To comprehend graphs as modeling and analysis tool
- To introduce various data structures with graph theory
- To learn a variety of different problems in graph theory
- To understand and analyze various graphs

Course Outcomes

Upon completion of the course, the students will be able to:

- Write mathematical definitions involving basic graphs
- Differentiate the potential use of directed and undirected graphs
- Develop algorithms based on diverse applications of graphs in different domains.
- Validate and critically assess a mathematical proof related with graphs

UNIT-I

BASICS OF GRAPHS AND TREES: Graphs – Introduction – Isomorphism – Sub Graphs – Walks, Paths, Circuits – Connectedness– Components – Euler Graphs – Hamiltonian paths and circuits – Trees – Properties of Trees– Distance and Centers in Tree – Rooted and Binary Trees.

UNIT-II

TREES, CONNECTIVITY & PLANARITY: Spanning Trees – Fundamental Circuits – Spanning Trees in a Weighted Graph – Cut Sets – Properties of Cut Set – All Cut Sets – Fundamental Circuits and Cut Sets – Connectivity and Separability – Combinational and Geometric Graphs – Planer Graphs – Different Representation of a Planer Graph.

UNIT-III

COLOURING AND DIRECTED GRAPH: Chromatic Number – Chromatic Partitioning – Chromatic Polynomial – Edge Coloring & Vertex Coloring –Vizing’s Theorem – Directed Graphs – Types of Directed Graphs – Digraphs and Binary Relations – Directed Paths and Connectedness – Euler Graphs.

UNIT-IV

MATCHINGS & COVERS: Matchings– Matchings& Coverings in Bipartite Graphs – Perfect Matching – Maximum Matching – Hall's Theorem & Consequences– Min – Max Theorems – Independent Sets & Edge Covers – Cuts & Connectivity

UNIT-V

PLANAR GRAPHS: Plane & Planar graphs – Dual Graphs – Euler Formula – Kuratowski's Theorem – The five-color theorem and four color conjecture.

Suggested Readings:

1. Douglas B. West, *Introduction to Graph Theory*, 2nd Edition, Prentice Hall of India,2015.
2. Narsingh Deo, *Graph Theory: With Application to Engineering and Computer Science*, 2nd Edition, Prentice Hall of India, 2003.
3. F. Harry, *Graph Theory*, Narosa Publications, 2001.
4. Rosen K.H., ―Discrete Mathematics and Its Applications, McGraw Hill, 2007.

Course Code	Course Title						Core/ Elective
PE 524 CS	**DATA ANALYTICS**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Any One Programming Language, Data Base & Basic Statistics	3	-	-	-	30	70	3

Course Objectives

- Overview of Data and Data analytics on huge datasets.
- Prepare Qualitative Data to perform different strategies of analytics
- Explore Data Analysis using R Software
- Able to realistically assess the application of data analytics technologies for different usage scenarios

Course Outcomes

Upon completion of the course, the students will be able to:

- Demonstrate proficiency with statistical analysis of data.
- Develop the ability to build and assess data-based models.
- Execute statistical analyses with professional statistical software.
- Demonstrate skill in data management.
- Apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively

UNIT - I

Getting to Know Your Data - Data Objects and Attribute Types - Attribute, Nominal Attributes, Binary Attributes, Ordinal Attributes, Numeric Attributes, Discrete versus Continuous Attributes**.** **Basic Measuring Data Similarity and Dissimilarity -** Data Matrix versus Dissimilarity Matrix, Proximity Measures for Nominal Attributes, Proximity Measures for Binary Attributes, Dissimilarity of Numeric Data: Minkowski Distance, Proximity Measures for Ordinal Attributes, Dissimilarity for Attributes of Mixed Types, Cosine Similarity**.**

UNIT - II

Introduction to Data Analytics - Big Data and Data Science, Small Data, A Short Taxonomy of Data Analytics, Examples of Data Use, Breast Cancer in Wisconsin, Polish Company Insolvency Data, A Little History on Methodologies for Data Analytics.

Descriptive Statistics - Scale Types, Descriptive Univariate Analysis, Univariate Frequencies, Contents, Univariate Data Visualization, Univariate Statistics, Common Univariate Probability Distributions, Descriptive Bivariate Analysis, Two Quantitative Attributes, Two Qualitative Attributes, at Least one of them Nominal, Two Ordinal Attributes.

UNIT - III

Descriptive Multivariate Analysis - Multivariate Frequencies, Multivariate Data Visualization, Multivariate Statistics, Location Multivariate Statistics, Dispersion Multivariate Statistics.

Data Quality and Preprocessing - Data Quality, Missing Values, Redundant Data, Inconsistent Data, Noisy Data, Outliers, Converting to a Different Scale Type, Converting Nominal to Relative, Converting Ordinal to Relative or Absolute, Converting Relative or Absolute to Ordinal or Nominal, Converting to a Different Scale.

UNIT - IV

Data Analytics Lifecycle Overview - Key Roles for a Successful Analytics Project, Background and Overview of Data Analytics Lifecycle - Discovery, Data Preparation, Model Planning, Model Building, Communicate Results, Operationalize and Case Study.
Data Analytics Methods using R - Introduction to R, R Graphical User Interfaces, Data Import and Export, Attribute and Data Types, Descriptive Statistics, Exploratory Data Analysis, Visualization Before Analysis, Dirty Data Visualizing a Single Variable Examining Multiple Variables, Data Exploration Versus Presentation.

UNIT - V

Data Visualization Basics - Key Points Supported with Data, Evolution of a Graph, Common Representation Methods, How to Clean Up a Graphic, Additional Considerations.

Applications of Data Analytics on Text & Web: Working with Texts, Data Acquisition, Feature Extraction, Tokenization, Stemming, Conversion to Structured Data, Trends, Sentiment Analysis, Web Mining, & Recommender Systems.

Suggested Text Books:

1. Data Mining: Concepts and Techniques Second Edition – Jiawei Han and Micheline Kamber – Morgan KaufMan Publisher, 2011
2. A General Introduction to Data Analytics, Joao Mendes Moreira,Andre C.P.L.F.de Carvalho, Tomas Horvath, Wiley Publications., 2018.
3. David Dietrich, Barry Hiller, “Data Science & Big Data Analytics”, EMC education services, Wiley publications, 2012.

Course Code	Course Title						Core/ Elective
PE 531 CS	**BLOCK CHAIN TECHNOLOGY**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- Understand how block chain systems (mainly Bitcoin and Ethereum) work,
- To securely interact with them,
- Design, build, and deploy smart contracts and distributed applications,
- Integrate ideas from block chain technology into their own projects.

Course Outcomes: Upon completion of the course, the students will be able to:

- Explain design principles of Bitcoin and Ethereum.
- Explain Nakamoto consensus.
- Explain the Simplified Payment Verification protocol.
- List and describe differences between proof-of-work and proof-of-stakeconsensus.
- Interact with a block chain system by sending and reading transactions.
- Design, build, and deploy a distributed application.
- Evaluate security, privacy, and efficiency of a given block chain system.

UNIT - I

Basics: Distributed Database, Two General Problem, Byzantine General problem and Fault Tolerance, Hadoop Distributed File System, Distributed Hash Table, ASIC resistance, Turing Complete.

Cryptography: Hash function, Digital Signature - ECDSA, Memory Hard Algorithm, Zero Knowledge Proof.

UNIT - II

Blockchain: Introduction, Advantage over conventional distributed database, Block chain Network, Mining Mechanism, Distributed Consensus, Merkle Patricia Tree, Gas Limit, Transactions and Fee, Anonymity, Reward, Chain Policy, Life of Block chain application, Soft & Hard Fork, Private and Public block chain.

UNIT - III

Distributed Consensus: Nakamoto consensus, Proof of Work, Proof of Stake, Proof of Burn, Difficulty Level, Sybil Attack, Energy utilization and alternate.

Unit - IV

Cryptocurrency: History, Distributed Ledger, Bitcoin protocols - Mining strategy and rewards, Ethereum -

Construction, DAO, Smart Contract, GHOST, Vulnerability, Attacks, Sidechain, Namecoin

Unit – V

Cryptocurrency Regulation: Stakeholders, Roots of Bit coin, Legal Aspects-Crypto currency Exchange, Black Market and Global Economy.

Applications: Internet of Things, Medical Record Management System, Domain Name Service and future of Block chain.

Case study : *Naive Blockchain construction, Memory Hard algorithm - Hashcash implementation, Direct Acyclic Graph, Play with Go-ethereum, Smart Contract Construction, Toy application using Blockchain, Mining puzzles*

Suggested Readings:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.
2. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System
3. DR. Gavin Wood, “ETHEREUM: A Secure Decentralized Transaction Ledger,” Yellow paper.2014.
4. Nicola Atzei, Massimo Bartoletti, and TizianaCimoli, A survey of attacks on Ethereum smart contracts

Course Code	Course Title						Core/ Elective
PE 532 CS	**INFORMATION RETRIEVAL SYSTEMS**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	-

Course Objectives

- To understand indexing and querying in information retrieval systems
- To learn the different models for information retrieval
- To expose the students to text classification and clustering
- To learn about web searching

Course Outcomes

- Understand the algorithms and techniques for information retrieval (document indexing and retrieval, query processing)
- Quantitatively evaluate information retrieval systems
- Classify and cluster documents
- Understand the practical aspects of information retrieval such as those in web search engines.

UNIT-I

Introduction to Information Retrieval Systems: Definition of Information Retrieval System, Objectives of Information Retrieval Systems, Functional Overview, Relationship to Database Management Systems, Digital Libraries and Data Warehouses.

Boolean Retrieval: An example information, Building an inverted index, processing Boolean queries, the extended Boolean model versus ranked retrieval.
The term vocabulary and postings lists: Document delineation and character sequence decoding, determining the vocabulary of terms, Faster postings list intersection via skip pointers, Positional postings, and Phrase queries.
Dictionaries and tolerant retrieval: Search structures for dictionaries, Wildcard queries, spelling correction.

UNIT-II

Index construction: Hardware basics, blocked sort-based indexing, Single-pass in-memory indexing, Distributed indexing, Dynamic indexing, Other types of indexes.
Index compression: Statistical properties of terms in information retrieval, Dictionary compression, Postings filecompression.
Cataloging and Indexing: History and Objectives of Indexing, Indexing Process, Automatic Indexing, Information Extraction.

Scoring, term weighting and the vector space model: Parametric and zone indexes, Term frequency and weighting, the vector space model for scoring, and Variant tf-idf functions.

UNIT-III

Evaluation in information retrieval: Information retrieval system evaluation, Standard test collections, Evaluation of unranked retrieval sets, Evaluation of ranked retrieval results, Assessing relevance.
Relevance feedback and query expansion: Relevance feedback and pseudo relevance feedback, Global methods for query reformulation.

Probabilistic information retrieval: Basic probability theory, The Probability Ranking Principle, The Binary Independence Model.

Language models for information retrieval: Language models, The query likelihood model.

UNIT-IV

Text classification and Naive Bayes: The text classification problem, Naive Bayes text classification, The Bernoulli model, Properties of Naive Bayes, and Feature selection.

Vector space classification: Document representations and measures of relatedness in vector spaces, Rocchio classification, k- nearest neighbour, Linear versus nonlinear classifiers.
Flat clustering: Clustering in information retrieval, Problem statement, Evaluation of clustering, k-means. **Hierarchical clustering:** Hierarchical agglomerative clustering, Single-link and complete-link clustering, Group-average agglomerative clustering, Centroid clustering, Divisive clustering.

UNIT-V

Matrix decompositions and Latent semantic indexing: Linear algebra review, Term-document matrices and singular value decompositions, Low-rank approximations, Latent semantic indexing.

Web search basics: Background and history, Web characteristics, Advertising as the economic model, The search user experience, Index size and estimation, Near-duplicates and shingling.
Web crawling and Indexes: Overview, Crawling, Distributing indexes, Connectivity servers.

Link analysis: The Web as a graph, Page Rank, Hubs and Authorities.

Suggested Readings:

1. Christopher D. Manning, PrabhakarRaghavan, HinrichSchütze, *An Introduction to Information Retrieval*, Cambridge University Press, Cambridge, England,2008
2. David A. Grossman, OphirFrieder, *Information Retrieval–Algorithms and Heuristics*, Springer, 2nd Edition (Distributed by Universities Press),2004.
3. Gerald J Kowalski, Mark T Maybury. *Information Storage and Retrieval Systems*, Springer,2000
4. SoumenChakrabarti, *Mining the Web: Discovering Knowledge from Hypertext Data*, Morgan- Kaufmann Publishers,2002.

Course Code	Course Title						Core/ Elective
PE 533 CS	**SOFT COMPUTING**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

Objectives of the course

- Classify the various soft computing frame works
- Be familiar with the design of neural networks, fuzzy logic and fuzzy systems
- Learn mathematical background for optimized genetic programming

Course Outcomes

Upon completion of the course, the students will be able to:

- Learn about soft computing techniques and their applications.
- Learn about fuzzy logic, various fuzzy systems and their functions.
- Use fuzzy rules and reasoning to develop decision making and expert system
- Choose and design suitable neural network for real time problems
- Understand the genetic algorithm concepts and their applications

UNIT-I

Introduction to Soft Computing: Soft computing constituents, characteristics of neuro-computing and soft computing, difference between hard computing and soft computing, some applications of soft computing techniques, concepts of learning and adaptation.

UNIT-II

Fuzzy logic: Introduction to classical/crisp sets and fuzzy sets, classical/crisp relations and fuzzy relations, tolerance and equivalence relations, non-iterative fuzzy sets.

Membership functions: fuzzification, methods of membership value assignments, defuzzification, lambda cuts for fuzzy sets and fuzzy relations, defuzzification methods.

UNIT-III

Fuzzy arithmetic and fuzzy measures: Fuzzy rule base and approximate reasoning, truth values and tables in fuzzy logic, fuzzy propositions, formation of rules, decomposition and aggregation of rules, fuzzy reasoning, fuzzy inference systems, fuzzy decision making, fuzzy logic control systems, fuzzy expert systems.

UNIT-IV

Introduction Neural Network: Fundamental concept, evolution of neural networks, models of artificial neural networks, important technologies, applications, McCulloch, Pitts Neuron, linear separability, Hebb network.

Supervised learning network: Perception networks, adaptive linear neuron, multiple adaptive linear neurons, back propagation network, radial basis function network.

Unsupervised learning networks: Kohonenself-organizing feature maps, learning vector quantization, counter propagation networks, adaptive resonance theory network.

UNIT-V

Genetic Algorithm: Difference between traditional algorithms and GA, genetic algorithm and search space, general genetic algorithm, operators, generational cycle, in genetic algorithm, stopping condition for genetic algorithm flow, constraints in genetic algorithm, schema theorem, classification of genetic algorithm, genetic programming, multilevel optimization.

Suggested Readings:

1. J.S.R.Jang, C.T.Sun and E.Mizutani, “Neuro-Fuzzy and Soft Computing”, Pearson Education 2004.
2. S.N.Sivanandam, S.N.Deepa “Principles of Soft Computing” Second Edition, Wiley Publication.
3. Timothy J.Ross, “Fuzzy Logic with Engineering Applications”, McGraw-Hill,1997.
4. Davis E.Goldberg, “Genetic Algorithms: Search, Optimization and Machine Learning”, Addison Wesley, N.Y.
5. S.Rajasekaran and G.A.VijayalakshmiPai “Neural Networks, Fuzzy Logic and Genetic Algorithms” PHI Learning.

Course Code	Course Title						Core/ Elective
PE 534 CS	**COMPUTER GRAPHICS**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Mathematics, Engg.Drawing	3	-	-	-	30	70	3

Course Objectives

- To introduce the concept of synthetic camera model , programmable pipeline and OpenGL API
- To study different interaction modes and data structures that store 2-D and 3-D geometric objects
- To understand different transformations in 2-D and 3-D
- To study different rasterization and rendering algorithms

Course Outcomes

After completing this course, the student will be able to:-

- Describe the steps in graphics programming pipeline
- Write interactive graphics applications using OpenGL geometric primitives
- Apply affine transformations for viewing and projections
- create realistic images of 3-d objects that involve lighting shading aspects

UNIT-I

Graphics Systems and Models: Graphics system, Images, Physical and Synthetic, Imaging system, Synthetic camera model, Programming interface, Graphics architectures, Programmable pipelines.

Graphics Programming: Programming two-dimensional applications, OpenGL API, Primitives and attributes, Color, Viewing and Control functions.

UNIT-II

Input and Interaction: Input devices, Display lists & modeling, Programming event-driven input, Picking, Building interactive models, Animating interactive programs, Logic operations.

Geometric Objects: Three-dimensional primitives, Coordinate systems and frames, Frames in OpenGL, Modeling colored cube.

UNIT-III

Transformations: Affine transformations, Transformations in homogeneous coordinates, Concatenation of transformations, OpenGL transformation matrices.

Viewing: Classical and Computer views, Viewing with a computer, Positioning of camera, Simple projections, Projections in OpenGL, Hidden surface removal, Parallel-projection matrices, Perspective-projection matrices.

UNIT-IV

Lighting and Shading: Light sources, The Phong lighting model, Computational vectors, Polygonal shading, Light sources in OpenGL, Specification of matrices in OpenGL, Global illumination.

From Vertices to Frames: Basic implementation strategies, Line-segment clipping, Polygon clipping, Clipping in three dimensions, Rasterization, Anti-aliasing.

UNIT-V

Modeling & Hierarchy: Hierarchal models, Trees and traversal, Use of tree data structure, Animation, Graphical objects, Scene graphs, Simple scene graph API, Open Scene graph, Other tree structures.

Suggested Reading

1. Edward Angel, Interactive Computer Graphics: A Top-Down Approach Using OpenGL, Pearson Education, 5th edition, 2009
2. Francis S Hill Jr., Stephen M Kelley, Computer Graphics using OpenGL, Prentice-Hall Inc., 3rd Edition, 2007
3. Jim X. Chen, Foundations of 3D Graphics Programming using JOGL and Java3D, Springer Verlag, 2006
4. Hearn Donald, Pauline M Baker, Computer Graphics, 2nd edition, 1995

CSE - SEMESTER - VI (Proposed for the academic year 2020-21)

S. No	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	D/P	t Hrs/W	CIE	SEE	Duration in Hrs/Wk	
Theory Course										
1.	PC 601 CS Core-10	Compiler Design	3	1	-	4	30	70	3	3
2.	PC 602 CS Core-11	Computer Networks	3	1	-	4	30	70	3	3
3.	PC 603 CS Core 12	Design and Analysis of Algorithms	3	1	-	3	30	70	3	3
4.	PE –IV	Professional Elective - IV	3	-	-	3	30	70	3	3
5	PE –V	Professional Elective -V	3	-	-	3	30	70	3	3
6	OE-I	Open Elective-I	3	-	-	-	30	70	3	3
Practical/Laboratory Course										
7	PC631 CS	Compiler Design Lab	-	-	2	2	25	50	3	1
8	PC632 CS	Computer Networks Lab	-	-	2	2	25	50	3	1
9	PC 633 CS	Design and Analysis of Algorithms Lab	-	-	2	2	25	50	3	1
10	SI 671 IT	Summer Internship*	-	-	-	-	-	-	-	-
Total			**18**	**3**	**6**	**27**	**205**	**570**		**21**

Profession Elective – IV	
Course Code	**Course Title**
PE 627CS	Advanced Operating Systems
PE 628 CS	Cloud Computing
PE 629 CS	Speech and Natural Language Processing
PE 630 CS	Machine Learning

Profession Elective – V	
Course Code	**Course Title**
PE 651 CS	Data Mining
PE 652CS	Human Computer Interaction
PE 653 CS	Digital Forensics
PE 654 CS	Internet of Things

Open Elective - I	
Course Code	**Course Title**
OE 601	Soft Skills & Interpersonal Skills
OE 602	Human Resource Development and Organizational Behaviour
OE 603	Cyber Law and Ethics

Course Code	Course Title						Core/ Elective
PC 601 CS	**COMPILER DESIGN**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	1	-	-	30	70	3

Course Objectives

- To understand and list the different stages in the process of compilation.
- Identify different methods of lexical analysis
- Design top-down and bottom-up parsers
- Identify synthesized and inherited attributes
- Develop syntax directed translation schemes
- Develop algorithms to generate code for a target machine

Course Outcomes

- Upon completion of the course, the students will be able to:
- For a given grammar specification, develop the lexical analyzer.
- For a given parser specification, design top-down and bottom-up parsers.
- Develop syntax directed translation schemes.
- Develop algorithms to generate code for target machine.

UNIT-I

Introduction: The Structure of a Compiler, Phases of Compilation, The Translation Process, Major Data Structures in a Compiler, Bootstrapping and Porting.

Lexical Analysis (Scanner): The Role of the Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, The Lexical Analyzer Generator Lex.

UNIT-II

Syntax Analysis (Parser): The Role of the Parser, Syntax Error Handling and Recovery, Top-Down Parsing, Bottom-Up Parsing, Simple LR Parsing, More Powerful LR Parsing, Using Ambiguous Grammars, Parser Generator Yaac.

UNIT-III

Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's Applications of Syntax-Directed Translation.

Symbol Table: Structure, Operations, Implementation and Management.

UNIT-IV

Intermediate Code Generation: Variants of Syntax Trees, Three-Address Code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Backpatching, Switch-statements, Intermediate Code for Procedures.

Run-time environment: Storage Organization, Stack Allocation of Space, Access to Nonlocal Data on the Stack, Parameter passing, Heap Management and Garbage Collection.

UNIT-V

Code Generation: Issues in the Design of a Code Generator, The Target Language, Addresses in the Target Code, Basic Blocks and Flow graphs, Optimization of Basic Blocks, Peephole Optimization, Register Allocation and Assignment.

Machine-Independent Optimizations: The Principal Sources of Optimizations, Introduction to Data-Flow Analysis.

Suggested Books:

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, & Jeffrey D. Ullman , *Compilers :Principles, Techniques and Tool**s***, 2nd Edition, Pearson Education, 2006.
2. Kenneth C. Louden**,** *Compiler Construction: Principles and Practice, Thomson Learning Inc., 1997.*
3. P.Trembley and P.S.Sorenson, ***The Theory and Practice of Compiler Writing***, TMH-1985.

Course Code	Course Title						Core/ Elective
PC 602 CS	**COMPUTER NETWORKS**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
	3	1	-	-	30	70	-

Course Objectives

- To develop an understanding of communication in modern network architectures from a design and performance perspective.
- To understand Data Transmission standards and MAC protocols.
- To introduce the protocols functionalities in Network Layer and Transport Layer.
- To understand DNS and supportive application protocols.
- To provide basic concepts of Cryptography.

Course Outcomes

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

- Explain the functions of the different layer of the OSI and TCP/IP Protocol.
- Understand wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) describe the function of each block.
- Illustrate network layer and transport layer protocols. For a given problem related TCP/IP protocol developed the network programming.
- Configure DNS , EMAIL, SNMP, Bluetooth, Firewalls using open source available software and tools.
- Identify the types of encryption techniques.

UNIT - I

Data communication Components: Representation of data communication, flow of Networks,Layered architecture, OSI and TCP/IP model, Transmission Media. (William stalling)

Techniques for Bandwidth utilization: Line configuration, Multiplexing - Frequency division, Time division and Wave division, Asynchronous and Synchronous transmission , XDSL , Introduction to Wired and Wireless LAN

UNIT - II

Data Link Layer and Medium Access Sub Layer: Error Detection and Error Correction -

Fundamentals, Block coding, Hamming Distance, CRC;

Flow Control and Error control protocols: Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking.

Multiple access protocols: Pure ALOHA, Slotted ALOHA, CSMA/CD,CDMA/CA

UNIT - III

Network Layer: Switching techniques (Circuit and Packet) concept ,**Logical addressing**: IPV4(Header), IPV6(Header), NAT , Sub-Netting concepts .

Inter-Networking: Tunnelling , Fragmentation , congestion control (Leaky Bucket and Token Bucket algorithm), Internet control protocols: ARP, RARP, BOOTP and DHCP.

Network Routing Algorithms: Delivery, Forwarding and Unicast Routing protocol, Gateway protocols.

UNIT - IV

Transport Layer: Process to Process Communication, Elements of transport protocol ,

Internet Transport Protocols: UDP, TCP.

Congestion and Quality of Service, QoS improving techniques.

UNIT - V

Application Layer: Domain Name Space (DNS), EMAIL, SNMP, Bluetooth.

Basic concepts of Cryptography: Network Security Attacks, firewalls, symmetric encryption, Data encryption Standards, public key Encryption (RSA), Hash function, Message authentication, Digital Signature.

Suggested books:

1. Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGrawHill.
2. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.
3. W. Richard Stevens, Unix Network Programming, Prentice Hall / Pearson Education, 2009

Course Code	Course Title						Core/ Elective
PC 603 CS	**DESIGN AND ANALYSIS OF ALGORITHMS**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Problem Solving Skills, Data Structures, Discrete Structures	3	-	-	-	30	70	-

Course Objectives

- Analyze the asymptotic performance of algorithms
- Write rigorous correctness proofs for algorithms
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis
- Synthesize efficient algorithms in common engineering design situations.

Course Outcomes

- Ability to analyze the performance of algorithms.
- Ability to choose appropriate algorithm design techniques for solving problems.
- Ability to understand how the choice of data structures and the algorithm design methods impact the performance of programs

UNIT-I

Introduction: Algorithm definition, and specification, asymptotic analysis – best, average, and worst-case behavior; Performance measurements of Algorithms, Time and Space complexities, Analysis of recursive algorithms.

Basic Data Structures: Disjoint set operations, union and find algorithms, Dictionaries, Graphs, Trees.

UNIT-II

Divide and Conquer: General method, Control abstraction, Merge sort, Quick Sort – Worst, Best and average case. Binary search.

Brute Force: Computing an– String Matching – Closest-Pair and Convex-Hull Problems - Exhaustive Search – Travelling Salesman Problem – Knapsack Problem – Assignment problem.

Greedy method: General method, applications- Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees, Single source shortest path problem.

UNIT-III

Dynamic Programming: General Method, applications- All pairs shortest path problem, Optimal binary search trees, 0/1 knapsack problem, Reliability design, Traveling sales person problem.

Backtracking: General method, Recursive backtracking algorithm, Iterative backtracking method. 8-Queen problem, Hamiltonian Cycle, 0/1 Knapsack Problem.

Branch and Bound: Control abstractions for Least Cost Search, Bounding, FIFO branch and bound, LC branch and bound, 0/1 Knapsack problem – LC branch and bound and FIFO branch and bound solution, Traveling sales person problem.

UNIT-IV

Graph Algorithms: Graph Traversals DFS, BFS, Transitive Closure, Directed Acyclic Graphs - Topological Ordering, Network Flow algorithms.
Tries: Standard Tries, Compressed Tries, Suffix Tries, Search Engine Indexing.
External Searching and B-Trees: (a, b) Trees, B-Trees

UNIT-V

Computational Complexity: Non Deterministic algorithms, The classes: P, NP, NP Complete, NP Hard, Satisfiability problem, Proofs for NP Complete Problems: Clique, Vertex Cover.
Parallel Algorithms: Introduction, models for parallel computing, computing with complete binary tree,

References:

1. E. Horowitz, S. Sahni, Fundamentals of Computer Algorithms.
2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, “Data Structures and Algorithms”, Pearson Education, Reprint 2006.
3. M T Goodrich, Roberto Tamassia, Algorithm Design, John Wiley, 2002.
4. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, “Introduction to Algorithms”, Third Edition, PHI Learning Private Limited, 2012.

Course Code	Course Title						Core/ Elective
PC 631 CS	**COMPILER DESIGN LAB**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	25	50	2

Course Objectives

- To learn usage of tools LEX, YAAC
- To develop a code generator
- To implement different code optimization schemes

Course Outcomes

- Generate scanner and parser from formal specification.
- Generate top down and bottom up parsing tables using Predictive parsing, SLR and LR Parsing techniques.
- Apply the knowledge of YACC to syntax directed translations for generating intermediate code – 3 address code.
- Build a code generator using different intermediate codes and optimize the target code.

List of Experiments to be performed:

1. Sample programs using LEX.
2. Scanner Generation using LEX.
3. Elimination of Left Recursion in a grammar.
4. Left Factoring a grammar.
5. Top down parsers.
6. Bottom up parsers.
7. Parser Generation using YACC.
8. Intermediate Code Generation.
9. Target Code Generation.
10. Code optimization.

Course Code	Course Title						Core/ Elective
PC 632 CS	**COMPUTER NETWORKS LAB**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
DC	-	-	-	2	30	70	-

Course Objectives

- Learn to communicate between two desktop computers.
- Learn to implement the different protocols
- Be familiar with socket programming.
- Be familiar with the various routing algorithms
- Be familiar with simulation tools.
- To use simulation tools to analyze the performance of various network protocols

Course Outcomes

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

- Implement various protocols using TCP and UDP.
- Program using sockets.
- Use simulation tools to analyze the performance of various network protocols.
- Implement and Analyze various routing algorithms.

1. Running and using services/commands like tcpdump, netstat, ifconfig, nslookup, FTP, TELNET and traceroute. Capture ping and trace route PDUs using a network protocol analyzer and examine.

2. Configuration of router, switch . (using real devices or simulators)

3. Socket programming using UDP and TCP (e.g., simple DNS, data & time client/server, echo client/server, iterative & concurrent servers)

4. Network packet analysis using tools like Wireshark, tcpdump, etc.

5. Network simulation using tools like Cisco Packet Tracer, NetSim, OMNeT++, NS2, NS3, etc.

6. Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS. Performance evaluation of Routing protocols using Simulation tools.

7. Programming using raw sockets

8. Programming using RPC

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

LABORATORY REQUIREMENT FOR STUDENTS:

HARDWARE:

1. Standalone desktops

SOFTWARE:

1. C / C++ / Java / Python / Equivalent Compiler
2. Network simulator like NS2/NS3/OPNET/ CISCO Packet Tracer / Equivalent

Course Code	Course Title						Core/ Elective
PC 633 CS	**DESIGN AND ANALYSIS OF ALGORITHMS LAB**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Problem Solving Skills, Data Structures, Discrete Structures	-	-	-	2	30	70	-

Course Objectives

- To learn the importance of designing an algorithm in an effective way by considering space and time complexity
- To learn graph search algorithms.
- To study network flow and linear programming problems
- To learn the dynamic programming design techniques.
- To develop recursive backtracking algorithms.

Course Outcomes

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

- Design an algorithm in a effective manner
- Apply iterative and recursive algorithms.
- Design iterative and recursive algorithms.
- Implement optimization algorithms for specific applications.
- Design optimization algorithms for specific applications.

1. Sort a given set of elements using the Quicksort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

2. Implement Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of

elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

3.Obtain the Topological ordering of vertices in a given digraph and

Compute the transitive closure of a given directed graph using Warshall's algorithm

4. Implement 0/1 Knapsack problem using Dynamic Programming.

5. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

6. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.

7. Print all the nodes reachable from a given starting node in a digraph using BFS method

and Check whether a given graph is connected or not using DFS method.

8. Find a subset of a given set S = {sl, s2,.....,sn} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions {1, 2, 6} and {1, 8}.A suitable message is to be displayed if the given problem instance doesn't have a solution

9. Implement any scheme to find the optimal solution for the Traveling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.

10. Find Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm.

11. Implement All-Pairs Shortest Paths Problem using Floyd's algorithm.

12. Implement N Queen's problem using Back Tracking.

Course Code	Course Title						Core/ Elective
SI 671 CS	**SUMMER INTERNSHIP**						**CORE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	50	-	2*

Course Objectives: To prepare the students

- To give an experience to the students in solving real life practical problems with all its constraints.
- To give an opportunity to integrate different aspects of learning with reference to real life problems.
- To enhance the confidence of the students while communicating with industry engineers and give an opportunity for useful interaction with them and familiarize with work culture and ethics of the industry.

Course Outcomes: On successful completion of this course student will be

- Able to design/develop a small and simple product in hardware or software.
- Able to complete the task or realize a pre-specified target, with limited scope, rather than taking up a complex task and leave it.
- Able to learn to find alternate viable solutions for a given problem and evaluate these alternatives with reference to pre-specified criteria.
- Able to implement the selected solution and document the same.

Summer Internship is introduced as part of the curricula for encouraging students to work on problems of interest to industries. A batch of two or three students will be attached to a person from an Industry / R & D Organization / National Laboratory for a period of 4 weeks. This will be during the summer vacation following the completion of the VI semester course. One faculty member will act as an internal guide for each batch to monitor the progress and interacts with the Industry guide.

After the completion of the project, students will submit a brief technical report on the project executed and present the work through a seminar talk to be organized by the department. Award of sessional marks are based on the performance of the student at the work place and awarded by industry guide and internal guide (25 Marks) followed by presentation before the committee constituted by the department (25 Marks). One faculty member will coordinate the overall activity of Summer Internship.

***Note:** * Students have to undergo summer internship of 4 weeks duration at the end of semester VI and credits will be awarded after evaluation in VII semester.*

Course Code	Course Title						Core/ Elective
PE 641 CS	**ADVANCED OPERATING SYSTEMS**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Operating System	3	-	-	-	30	70	3

Course Objectives

- To study, learn, and understand the main concepts of advanced operating systems (parallel processing systems, distributed systems, real time systems, network operating systems, and open source operating systems);
- To learn hardware and software features that support these systems.

Course Outcomes

Upon completion of the course, the students will be able to:

- Understand the design approaches of advanced operating systems
- Analyse the design issues of distributed operating systems.
- Evaluate design issues of multiprocessor operating systems.
- Identify the requirements of database operating systems.
- Formulate the solutions to schedule the real time applications

UNIT-I

Architecture of Distributed Systems: Types, Distributed Operating System, Issues in Distributed Operating Systems, Theoretical Foundations: Global Clock, Lamport's Logical Clock, Vector Clocks, Global State, and Termination Detection.

UNIT-II

Distributed Mutual Exclusion: Classification, requirement, performance, non-token based algorithms, Lamport's algorithm, the Richart-Agarwala algorithm, token-based algorithm-Suzuki Kasami's broadcast algorithm, Singhal’s heuristic algorithm.

Deadlock Detection: Resource vs Communication deadlock, A graph- theoretic model, prevention, avoidance, detection, control organization, centralized deadlock-detection algorithm, the completely centralized algorithm, the HO-Ramamoorthy algorithm. Distributed deadlock detection algorithm - path - pushing, edge-chasing, hierarchical deadlock detection algorithm, menace-muntz and Ho-Ramamoorthy algorithm. Agreement Protocols: The system model, the Byzantine agreement, and the consensus problem.

UNIT-III

Distributed File System: Mechanisms, Design Issues, ***Case Studies:*** *Sun NFS, Sprite File System, DOMAIN, Coda File System.*

Distributed Shared Memory: Algorithms for Implementing DSM, Memory Coherence, Coherence Protocols, Design Issues, ***Case Studies:*** *IVY, Mirage, Clouds.*

Distributed Scheduling: Issues in Load Distribution, Components of Algorithm, Stability Load Distributing Algorithm, Performance.

UNIT-IV

Failure Recovery: Backward, Forward Error Recovery in Concurrent Systems, Consistent Set of Checkpoints, Synchronous and Asynchronous Checkpointing and Recovery.

Fault Tolerance: Commit Protocols, Non-Blocking Commit Protocols, Voting Protocols.

Protection and Security: Access Matrix, Private Key, Public key, and Kerberos System.

UNIT-V

Multiprocessor Operating Systems: Motivation, Basic Multiprocessor System Architecture, Interconnection Networks for Multiprocessor Systems, Caching, Hypercube Architecture. Threads, Process Synchronization, Processor Scheduling, and Memory Management.

Database Operating System: Concurrence Control, Distributed Databases, and Concurrency Control Algorithms.

Suggested Readings:

1. Singhal M, Shivaratri N.G, Advanced Concepts in Operating Systems, McGraw-Hill Intl., 1994.
2. Pradeep K Sinha, Distributed Operating Systems Concepts and Design, PHI, First Edition, 2002.
3. Andrew S. Tanenbaum, Distributed Operating Systems, Pearson Education India, First Edition, 2011.

Course Code	Course Title						Core/ Elective
PE 642 CS	**CLOUD COMPUTING**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To understand the concept of cloud computing.
- To understand the various issues in cloud computing.
- To familiarize themselves with the lead players in cloud.
- To appreciate the emergence of cloud as the next generation computing paradigm.

Course Outcomes

- Articulate the main concepts, key technologies, strengths and limitations of cloud computing. Identify the architecture, infrastructure and delivery models of cloud computing.
- Explain the core issues of cloud computing such as security, privacy and interoperability.
- illustrate the use of various cloud services available online

UNIT-I

INTRODUCTION - Historical Development - Cloud Computing Architecture – The Cloud Reference Model – Cloud Characteristics – Cloud Deployment Models: Public, Private, Community, Hybrid Clouds - Cloud Delivery Models: IaaS, PaaS, SaaS.

UNIT-II

CLOUD COMPUTING MECHANISM: Cloud Infrastructure Mechanism: Cloud Storage, Cloud Usage Monitor, Resource Replication – Specialized Cloud Mechanism: Load Balancer, SLA Monitor, Hypervisor, Resource Cluster, Multi Device Broker,

UNIT-III

STATE MANAGEMENT DATABASE – Cloud Management Mechanism: Remote Administration System, Resource Management System, SLA Management System, State Management Database – Cloud Management Mechanism: Remote Administration System, Resource Management System, SLA Management System,.

UNIT-IV

SECURITY IN THE CLOUD: Basic Terms and Concepts – Threat Agents – Cloud Security Threats – Cloud Security Mechanism: Encryption, Hashing, Digital Signature, Public Key Infrastructure, Identity and Access Management. Data Security :Application Security –Virtual Machine Security .

UNIT-V

CASE STUDIES :Google App Engine(GAE) – GAE Architecture – Functional Modules of GAE – Amazon Web Services(AWS) – GAE Applications – Cloud Software Environments – Eucalyptus – Open Nebula – Open Stack

Suggested Readings:

1. Thomas Erl, ZaighamMahood, Ricardo Puttini, ―Cloud Computing, Concept, Technology and Architecture‖, Prentice Hall, 2013.
2. Toby Velte, Anthony Velte, Robert C. Elsenpeter, ―Cloud Computing, A Practical Approach‖, Tata McGraw-Hill Edition, 2010.
3. Rittinghouse, John W., and James F. Ransome, "Cloud Computing: Implementation, Management, And Security", CRC Press, 2017.

Course Code	Course Title						Core/ Elective
PE 629 CS	**SPEECH AND NATURAL LANGUAGE PROCESSING**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- Teach students the leading trends and systems in natural language processing.
- Make them understand the concepts of morphology, syntax and semantics of the language and that they are able to give the appropriate examples that will illustrate the above mentioned concepts.
- Teach them to recognize the significance of pragmatics for natural language understanding.
- Enable students to be capable to describe the application based on natural language processing and to show the points of syntactic and semantic processing.

Course Outcomes

- To tag a given text with basic Language features
- To design an innovative application using NLP components
- To implement a rule based system to tackle morphology/syntax of a language
- To design a tag set to be used for statistical processing for real-time applications
- To compare and contrast the use of different statistical approaches for different types of NLP applications.
- Perform various language phonetic analysis

UNIT I

Introduction of NLP: Origins and challenges of NLP, Language Modeling: Grammar-based LM,
Statistical LM – Regular Expressions, Automata , Morphology and Finite State Transducers, Tokenization, stemming, Normalization, Detecting and Correcting Spelling Errors, Minimum Edit
Distance.

UNIT II

WORD LEVEL ANALYSIS: N-grams, Evaluating N-grams, Smoothing,Word Classes, Part-of-Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in PoS tagging – Entropy, Hidden Markov and Maximum Entropy models, ; Named Entities

UNIT-III

SYNTACTIC ANALYSIS: Context free rules and trees – The noun Phrase – Co-ordination – Verb
phrase – context free grammars – Parsing with context free grammars, Shallow parsing – Probabilistic CFG , Dependency Grammar , Semantic Analysis: Meaning Representation-

Lexical
Semantics- Ambiguity-Word Sense Disambiguation. Discourse Processing: cohesion- Reference
Resolution- Discourse Coherence and Structure.

UNIT_IV
Speech Fundamentals: Phonetics – speech sounds and phonetic transcription – articulatory phonetics – phonological categories and pronunciation variation – acoustic phonetics and signals –
phonetic resources – articulatory and gestural phonology

UNIT-V
Speech synthesis – text normalization – phonetic analysis – prosodic analysis – diphone waveform
synthesis – unit selection waveform synthesis – evaluation

Text Books:

1. Daniel Jurafsky, James H. Martin**—**Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, Pearson Publication, 2014.
2. Steven Bird, Ewan Klein and Edward Loper, **—**Natural Language Processing with Python, First Edition, OReilly Media, 2009.

Course Code	Course Title						Core/ Elective
PE 630 CS	**MACHINE LEARNING**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To learn the concept of how to learn patterns and concepts from data correlation.
- To design and analyze various machine learning algorithms and techniques with a modern outlook focusing on recent advances.
- Explore supervised and unsupervised learning paradigms of machine learning.
- To explore Deep learning technique and various feature extraction strategies.

Course Outcomes

Upon completion of the course, the students will be able to:

- Extract features that can be used for a particular machine learning approach in various applications.
- To compare and contrast pros and cons of various machine learning techniques and to get an insight of when to apply a particular machine learning approach.
- To mathematically analyze various machine learning approaches and paradigms.

UNIT-I

Supervised Learning (Regression/Classification) - Basic methods: Distance-based methods, Nearest-Neighbours, Decision Trees, Naive Bayes, Linear models: Linear Regression, Logistic Regression, Generalized Linear Models, Support Vector Machines, Nonlinearity and Kernel Methods, Beyond Binary.

Classification: Multi-class/Structured Outputs, Ranking.

UNIT-II

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

UNIT-III

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

UNIT-IV

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

UNIT-V

Scalable Machine Learning (Online and Distributed Learning) A selection from some other advanced topics, e.g., Semi-supervised Learning, Active Learning, Reinforcement Learning, Inference in Graphical Models, Introduction to Bayesian Learning and Inference. Recent trends in various learning techniques of machine learning and classification methods for IOT applications. Various models for IOT applications.

Suggested Readings:

1. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012
2. Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, The Elements of Statistical Learning, Springer 2009 (freely available online)
3. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2007.

Course Code	Course Title						Core/ Elective
PE 651 CS	**DATA MINING**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To introduce the basic concepts of data Mining and its applications
- To understand different data mining like classification, clustering and Frequent Pattern mining
- To introduce current trends in data mining
- To understand, pre-process and analyze the basic concepts of Data Attributes
- To explore the various data mining techniques (Association Analysis, Classification, Clustering) adapted on data as per the requirement

Course Outcomes

- Organize and Prepare the data needed for data mining using preprocessing techniques
- Implement the appropriate data mining methods like classification, clustering or Frequent Pattern mining on a given data set
- Define and apply metrics to measure the performance of various data mining algorithms
- Understanding the importance of data mining application and using the most appropriate approach or trend for the realistic strategy

UNIT-I

INTRODUCTION: What is Data Mining? The process of knowledge discovery in databases, predictive and descriptive data mining techniques, supervised and unsupervised learning techniques. Major issues in Data Mining. Getting to know your data: Data objects and attributed types. Basic statistical descriptions of data. Data visualization, Measuring data similarity and dissimilarity.

UNIT-II

MINING FREQUENT PATTERNS, ASSOCIATIONS AND CORRELATIONS: Basic concepts and methods, Frequent Item set Mining Methods, Sequential Pattern Mining concepts and Pattern evaluation methods.

UNIT-III

CLASSIFICATION: Basic concepts, Decision tree, Decision rules, Bayes classification methods, Advance methods, Bayesian Belief Network, K-Nearest Neighbor (KNN) classifier, Classification by back propagation, Support vector machine.

UNIT-IV

CLUSTER ANALYSIS: Concepts and Methods: Type of data in Cluster Analysis, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid-Based Methods, Evaluation of clustering.

UNIT-V

DATA MINING TRENDS AND RESEARCH FRONTIERS: Mining Complex Data Types, Other Methodologies of Data Mining, Data Mining Applications, Data Mining and Society, Data Mining trends.

Suggested Readings:

1. Jiawei Han, MichelineKamber, Jin Pei, Data Mining: Concepts & Techniques, 3rd Edition., MorgonKoffman,2011
2. VikramPudi, P. Radha Krishna, Data Mining, Oxford University Press, 1st Edition,2009.
3. Pang-Ning Tan, Michael Steinbach, AKarpatne, and Vipin Kumar, Introduction to Data Mining, 2nd Ed., Pearson Education, 2018.
4. J Zaki Mohammed and Wagner Meira, Data Mining and Analysis: Fundamental Concepts and Algorithms,Cambridge University Press,2014

Course Code	Course Title						Core/ Elective
PE 652 CS	**HUMAN COMPUTER INTERACTION**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- Learn the foundations of Human Computer Interaction
- Be familiar with the design technologies for individuals and persons with disabilities
- Be aware of mobile human computer interaction
- Learn the guidelines for human interface

Course Outcomes

- Understand the structure of models and theories of Human Computer Interaction and Vision
- Design an interactive Web interface on the basis of model studied

UNIT- I

Human: I/O Channels – Memory- Reasoning and Problem Solving;

Interaction: Models –Frameworks –Ergonomics- styles – elements – interactivity-

paradigms

Interactive Design Basics – process-scenarios-navigation-screen design –iteration and prototyping

UNIT- II

HCI in software process – usability engineering – prototyping in practice – design rationale

Design rules – principles, standards, guidelines, rules,

Evaluation techniques- Universal design

UNIT-III

Cognitive models – Socio-Organizational issues and stake holder requirements

Communication and collaboration models – Hypertext, Multimedia and WWW

UNIT- IV

Mobile Ecosystem: platforms, Application frameworks –

Types of mobile applications: Widgets, applications, Games - Mobile information architecture,

Mobile 2.0, Mobile Design: elements of mobile design, tools,

UNIT- V

Design of Web interfaces – Drag and Drop, Direct selection, Contextual tools,

Overlays, inlays and virtual pages, process flow, case studies,

Recent trends: Speech recognition and translation, multimodal system

References:

1. Alan Dix, Janet Finlay, Gregory Abowd, Russel Beale," Human Computer Interaction", 3rd Edition, Pearson Education 2004
2. Brain Fling, "Mobile Design and Development" First edition Orielly Media Inc. 2009
3. Bill Scott and Theresa Neil, "Designing Web Interfaces", First edition, Orielly 2009

Course Code	Course Title						Core/ Elective
PE 653 CS	**DIGITAL FORENSICS**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	30	70	3

Course Objectives

- To understand the basic digital forensics and techniques for conducting the forensic examination on different digital devices.
- To understand how to examine digital evidences such as the data acquisition, identification analysis.

Course Outcomes

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

- Apply forensic analysis tools to recover important evidence for identifying computer crime.
- Be well-trained as next-generation computer crime investigators.

UNIT -I

Computer forensics fundamentals, Benefits of forensics, computer crimes, computer forensics evidence and courts, legal concerns and private issues.

UNIT- II

Understanding Computing Investigations – Procedure for corporate High-Tech investigations, understanding data recovery work station and software, conducting and investigations.

UNIT-III

Data acquisition- understanding storage formats and digital evidence, determining the best acquisition method, acquisition tools, validating data acquisitions, performing RAID data acquisitions, remote network acquisition tools, other forensics acquisitions tools.

UNIT-IV

Processing crimes and incident scenes, securing a computer incident or crime, seizing digital evidence at scene, storing digital evidence, obtaining digital hash, reviewing case.

UNIT-V

Current computer forensics tools- software, hardware tools, validating and testing forensic software, addressing data-hiding techniques, performing remote acquisitions, E-Mail investigations- investigating email crime and violations, understanding E-Mail servers, specialized E-Mail forensics tool.

Text Books

1. Warren G. Kruse II and Jay G. Heiser, “Computer Forensics: Incident Response Essentials”, Addison Wesley, 2002.
2. Nelson, B, Phillips, A, Enfinger, F, Stuart, C., “Guide to Computer Forensics and Investigations, 2nd ed., Thomson Course Technology, 2006.
3. Vacca, J, Computer Forensics, Computer Crime Scene Investigation, 2nd Edition, Charles River Media, 2005.

Course Code	Course Title						Core/ Elective
PE 654 CS	**INTERNET OF THINGS**						**ELECTIVE**
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
Programming in C, OS, CN, WT	3	-	-	-	30	70	3

Course Objectives

Students understanding will be enhanced by:-

- Exploration towards the integration of the physical and logical worlds
- Exposure in understanding how IoT devices are designed & developed

Course Outcomes

After completing this course, the student will be able to:-

- Able to understand the application areas of IOT
- Able to realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks
- Able to understand building blocks of Internet of Things and characteristics

UNIT I

Introduction & Concepts: Introduction to Internet of Things (IoT), Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, IoT Levels

UNIT II

Architecture of IoT, Taxonomy, Sensors and Actuators, Preprocessing, Communication, Middleware, Applications of IoT

UNIT III

Introduction to ARDUINO: Getting Started with ARDUINO products, Built-In Examples

ARDUINO IoT Cloud: ARDUINO IoT Cloud Components

UNIT IV

Developing Internet of Things & Logical Design using Python: Introduction, IoT Design Methodology.

Basics of Python: Python Data Types & Data Structures, Control Flow, Functions, Modules, Packages, File Handling, Date/ Time Operations, Classes.

UNIT V

IoT Physical Devices & Endpoints: What is an IOT Device, Exemplary Device, Board, Linux on Raspberry Pi,

Interfaces, and Programming & IOT Devices.

Suggested Reading

1. Adrian McEwen, “Designing the Internet of Things”, Wiley Publishers, 2013.