NRI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution Permanently Affiliated to JNTUK, Kakinada)
(Accredited by NAAC with "A" Grade and ISO 9001:2015 Certified Institution)
POTHAVARAPPADU (V), (VIA) NUNNA, AGIRIPALLI (M), PIN – 521 212

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE STRUCTURE FOR THIRD YEAR B.TECH PROGRAMME

III YEAR I SEMESTER

Sl. No	Course Code	Title of the Course	Scheme of Instruction (Periods Per Week)				Scheme of Examination (Maximum Marks)			No. of Credits
			L	T	P/D	Total	CIA	SEA	Total	
1	18A3105401	Advanced Data Structures	2	1	0	3	40	60	100	3
2	18A3105402	Computer Networks	2	1	0	3	40	60	100	3
3	18A3105403	Formal Languages and Automata Theory	3	0	0	3	40	60	100	3
4	PE-I	**Professional Elective-1**	3	0	0	3	40	60	100	3
5	18A3105601 18A3105602	**Open Elective-II** 1. Web Technology	3	0	0	3	40	60	100	3
6	18A3105492	Computer Networks lab	0	1	2	3	40	60	100	1.5
7	18A3105491	ADS Lab	0	0	3	2	40	60	100	1.5
8	18A3101301	Basics of Civil and Mechanical Engineering / Swayam/NPTEL **	0	2	0	2	40	60*	100	1
9	18A3105791	Competitive Coding	0	0	2	2	40	60*	100	1
10	18A3105792	Seminar	0	0	2	2	40	60*	100	1
11	18A3100802	Indian Constitution	0	1	2	2	40	60*	100	0
Total			**13**	**6**	**4**	**23**	**440**	**660**	**1100**	**21**

Code	Professional Elective – 1
18A3105511	1.1 Scripting Languages
18A3105512	1.2 Computer Graphics and Multimedia animation
18A3105513	1.3 Data Warehousing and Data Mining
18A3105514	1.4 Principles of Programming Languages

* No External Evaluation

Students can opt any one of the self-learning courses from approved online platforms viz., Swayam/NPTEL/MOOCS and produce the course completion certificate before the commencement of III-II Academic session. **The course has to be completed in a stipulated time.

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE STRUCTURE FOR THIRD YEAR B.TECH PROGRAMME

III YEAR II SEMESTER

Sl. No	Course Code	Title of the Course	Scheme of Instruction (Periods Per Week)				Scheme of Examination (Maximum Marks)			No. of Credits
			L	T	P/D	Total	CIA	SEA	Total	
1	18A3205401	Operating Systems	2	1	0	3	40	60	100	3
2	18A3205402	Design and Analysis of Algorithms	3	0	0	3	40	60	100	3
3	18A3205403	Compiler Design	3	0	0	3	40	60	100	3
4	PE-2	**Professional Elective -2**	3	0	0	3	40	60	100	3
5	PE-3	**Professional Elective -3**	3	0	0	3	40	60	100	3
6	18A3205404	Artificial Intelligence	3	0	0	3	40	60	100	3
7	18A3205491	Operating Systems & Unix programming Lab	0	0	2	2	40	60	100	1
8	18A3205492	R Programming lab	0	0	2	2	40	60	100	1
9	18A3205991	Aptitude and Reasoning – 2	0	0	2	2	40	60*	100	1
10	18A3205791	Hackathon	0	0	2	2	40	60*	100	1
11	18A3200791	Biology for Engineers/ Enterprising and Startup/ NSS / YOGA / Social service/ sports /games	0	2	0	2	40	60*	100	0
Total			**17**	**3**	**8**	**28**	**440**	**660**	**1100**	**22**

Code	Professional Elective -2	Code	Professional Elective -3
18A3205511	2.1 Advanced Database Management Systems	18A3205521	3.1 Computer Vision
18A3205512	2.2 UML & Design Patterns	18A3205522	3.2 Data Analytics
18A3205513	2.3 Distributed Systems	18A3205523	3.3 Software Testing Methodologies
18A3205514	2.4 Adhoc and Sensor Networks	18A3205524	3.4 Cloud Computing and Application Development

* No External Evaluation

III - I Courses

Course Code-Advanced Data Structures

Lecture – Tutorial-Practical::	2-1-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites: Data Structures, Programming with C

Advance Data Structures

Course Objectives:

Describe and implement a variety of advanced data structures (hash tables, priority queues, balanced search trees, digital search trees).

Analyze the space and time complexity of the algorithms studied in the course.

Identify different solutions for a given problem; analyze advantages and disadvantages to different solutions.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Able to understand the importance and application of Hashing
CO2	Able to understand skip lists
CO3	To get a good understanding about different balanced trees.
CO4	Be able to understand heaps and binomial queues.
CO5	Have an idea of applications of algorithms in a variety of areas, like string matching, indexing etc
CO6	Able to understand different search trees

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	1	2	2	-	2	-	-	-	-	-	-	-
CO2	1	1	1	2	1	-	-	-	-	-	-	-
CO3	1	2	3	2	1	-	-	-	-	-	-	-
CO4	2	3	2	2	1	-	-	-	-	-	-	2
CO5	1	2	-	3	3	-	-	-	-	-	-	-
CO6	-	2	3	3	-	-	-	-	-	-	-	2

UNIT I : Hashing and Skip Lists

Static Hashing- Hash Table, Hash Functions- Secure Hash Function, Overflow Handling, Theoretical Evaluation of Overflow Techniques, Dynamic Hashing- Motivation for Dynamic Hashing, Dynamic Hashing Using Directories, Directory less Dynamic Hashing, Hash Table Restructuring, Skip Lists, Analysis of Skip Lists.

UNIT II: Balanced Trees

AVL Trees: Maximum Height of an AVL Tree, Insertions and Deletions. 2-3 Trees : Insertion, Deletion, applications, introduction to Red-black trees

UNIT III: Priority Queues

Binary Heaps : Implementation of Insert and Delete min, Creating Heap.

Binomial Queues : Binomial Queue Operations, Binomial Amortized Analysis, Lazy Binomial Queues

UNIT IV: Pattern matching and Tries

Pattern matching algorithms- the Boyer –Moore algorithm, the Knuth-Morris-Pratt algorithm

Tries: Definitions and concepts of digital search tree, Binary trie, Patricia , Multi-way trie

TEXT BOOKS:

1. Fundamentals of DATA STRUCTURES in C: 2 nd ed, , Horowitz , Sahani, Anderson-freed, Universities Press.
2. Data structures and Algorithm Analysis in C, 2 nd edition, Mark Allen Weiss, Pearson

REFERENCE BOOKS:

1. Data Structures, a Pseudocode Approach, Richard F Gilberg, Behrouz A Forouzan,Cengage.
2. Introduction to Algorithms, 3rd Edition by *Thomas H.* Cormen , *Charles E. Leiserson, Ronald L. Rivest, Clifford Stein*

E-RESOURCES:

1. Web : http://lcm.csa.iisc.ernet.in/dsa/dsa.html
2. http://utubersity.com/?page_id=878
3.http://freevideolectures.com/Course/2519/C-Programming-and-Data-Structures
4.http://freevideolectures.com/Course/2279/Data-Structures-And-Algorithms

Course Code- COMPUTER NETWORKS

Lecture – Tutorial-Practical::	2-1-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites: Basics of Data Communication

Course Objectives:

1. Understand state-of-the-art in network protocols, architectures, and applications.
2. Process of networking research
3. Constraints and thought processes for networking research
4. Problem Formulation—Approach---Analysis---

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Understand OSI and TCP/IP models.
CO2	Design applications using internet protocols.
CO3	Understand routing and congestion control algorithms.
CO4	Understand how the internet works.

Contribution of Course Outcomes towards the achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	1	2	1	2	2				
CO2	3	3	3	2	3	2	2	2				
CO3	3	3	3	2	2	2	2	2				
CO4	3	2	2	2	2	1	2	2				

UNIT I: Introduction

Introduction to Computer Networks, Network Hardware, and Software, Network Topologies, The OSI Reference Model, The TCP/IP Reference Model, A Comparison of the OSI and TCP/IP Reference Models.

UNIT II: The Data Link Layer

The Data Link Layer Design Issues, Error Detection, and Correction- Error-Correcting Codes, Error Detecting Codes, Elementary Data Link Protocols, Sliding Window Protocols, Example Data Link Protocols.

MAC Sub-Layer: Channel Allocation Methods, Multiple Access Protocols, CSMA Protocols, Collision Free Protocols

UNIT III: The Network Layer

The Network Layer Design Issues, Routing Algorithms- The Optimality Principle, Shortest Path, Distance Vector Algorithms, Congestion Control Algorithms- Approaches to Congestion Control, Traffic-Aware Routing, Admission Control, Traffic Throttling, Load Shedding, The Network Layer in the Internet- IP Addresses, The IP Version 4, 6 Protocols.

UNIT IV: The Transport And Application Layers

Transport Layer- The Internet Transport Protocols- TCP, UDP.

Application Layer-The Domain Name System, Electronic Mail.

TEXT BOOKS:

1. Tanenbaum and David J Wetherall, Computer Networks, 5th Edition, Pearson Edu, 2010.

REFERENCE BOOKS:

1. Computer Networks: A Top-Down Approach, Behrouz A. Forouzan, FirouzMosharraf, McGraw Hill Education.
2. Computer Networks, 5ed, David Patterson, Elsevier.
1. Larry L. Peterson and Bruce S. Davie, “Computer Networks- A Systems Approach” 5th Edition, Morgan Kaufmann/Elsevier, 2011.
2. Computer Networks, Mayank Dave, CENGAGE.
3. An Engineering Approach to Computer Networks-S.Keshav, 2nd Edition, Pearson Education.
4. Understanding communications and Networks, 3rd Edition, W.A. Shay, Thomson

E-RESOURCES:

http://www.uoitc.edu.iq/images/documents/informatics-institute/exam_materials/Computer%20Networks%20-%20A%20Tanenbaum%20-%205th%20edition.pdf
https://www.slideshare.net/pawan1809/computer-networks-a-tanenbaum-5th-editionee
https://www.technolamp.co.in/2010/08/computer-networks-tanenbaum-powerpoint.html
http://citengg.blogspot.com/p/behrouz-forouzancomputer-networks4th.html
https://www.youtube.com/watch?v=O--rkQNKqls&list=PLbRMhDVUMngf-peFloB7kyiA40EptH1up

Course Code- Formal Languages and Automata Theory

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites: The students are expected to have a strong background in the fundamentals of discrete mathematics (symbolic logic, set, induction, number theory, summation, series, combinatorics, graph, recursion, basic proof techniques, etc.), algorithms and data structures. Some knowledge of programming languages, programming, and computer architecture will be helpful

THEORETICAL FOUNDATIONS OF COMPUTATION

Course Objectives:

1. To give an overview of the theoretical foundations of computer science from the perspective of formal
   languages
2. To illustrate finite state machines to solve problems in computing
3. To explain the hierarchy of problems arising in the computer sciences.
4. To familiarize Regular grammars, context frees grammar.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	To use basic concepts of formal languages of finite automata techniques
CO2	To Design Finite Automata's for different Regular Expressions and Languages
CO3	To Construct context free grammar for various languages
CO4	To solve various problems of applying normal form techniques, push down automata and Turing Machines
CO5	To participate in GATE, PGECET and other competitive examinations

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	3	1	1	-	2	1	1	-	1	-	2
CO2	2	2	3	2	2	2	1	-	1	2	1	2
CO3	2	3	3	2	2	2	-	-	1	1	1	2
CO4	3	2	2	2	3	1	1	1	1	1	1	1
CO5	3	3	3	-	-	-	-	-	-	-	-	2

UNIT I :

FINITE AUTOMATA (FA): Introduction to Finite Automata, Central Concepts of Automata Theory, Deterministic Finite Automata (DFA), Nondeterministic Finite Automata (NFA), Equivalence of NFA and DFA. Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata, Finite automata with output (Moore and Mealy machines) and Inter conversion. Text search using automata.

UNIT II:

REGULAR EXPRESSIONS (RE): Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA's to Regular Expressions, Converting Regular Expressions to Automata, applications of Regular Expressions.

REGULAR GRAMMARS: Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular -Pumping lemma, Closure properties of regular languages. Applications of RE – RE in Unix

UNIT III:

CONTEXTS FREE GRAMMER (CFG): Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFG's, Minimization of CFG's, CNF, GNF, Pumping Lemma for CFL's, Enumeration of Properties of CFL (Proof's omitted). Applications of CFG

UNIT IV:

Push Down Automata(PDA)

Languages of PDA Acceptance by final state, empty stack and conversion. Equivalence of PDA's and CFG's, Deterministic Push Down Automata.

Turing Machines(TM)

Definition, Notation, Instantaneous Description and Languages, Design of TM's, Extension of Basic TM: Multitape TM, Restricted TM, Semi-infinite tape Machines, Multi-stack TM. Introduction to undecidable problems, Post-correspondence Problem.

TEXT BOOKS:

1. "Introduction to Automata Theory, Languages and Computation" by John E. Hop croft, Rajeev Motwani,
   Jeffrey D.Ullman: 3rd Edition, Pearson education, 2007.

REFERENCE BOOKS:

1. K. L. P Mishra, N. Chandrashekaran (2003), Theory of Computer Science-Automata Languages and Computation, 2nd edition, Prentice Hall of India, India.
2. Introduction to Languages and Automata Theory" by John C Martin, 3rd Ed, Tata McGraw-Hill, 2007.
3. "An Introduction to formal Languages and Automata" by Peter Linz,Narosa publishing house, II edition, 1997

E-RESOURCES:

1. Foundations of Computation-CAROL CRITCHLOW, DAVID ECK
2. Introduction to Theory of Computation- Anil Maheshwari,Michielsmid-carleton University-2012

MOOCs:

1. www.nptel/videos.in/2012/11/theory-of-computation.html
2. nptel.ac.in/courses/106104028/theory of computation.

Course Code- Scripting languages

Type of Course : Professional Elective – 1.1

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites:

Computer Programming, Web Technologies

Course Objectives:

- The course demonstrates an in depth understanding of the tools and the scripting languages necessary for design and development of applications dealing with Bio- information/ Bio-data.
- The instructor is advised to discuss examples in the context of Bio-data/ Bio-information application development.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Ability to understand the differences between scripting languages
CO2	Ability to apply your knowledge of the weaknesses of scripting languages to select implementation.
CO3	Able to gain some fluency programming in Perl and related languages.
CO4	Identify PHP encryption functions and Mcrypt Package.
CO5	Understand PHP Authentication and Methodologies

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2- Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	2										2
CO2	3	3	2									
CO3	3		2									2
CO4			3		3							2
CO5	3		3		2							2

UNIT I :

Introduction to PERL and Scripting Scripts and Programs, Origin of Scripting , Scripting Today, Characteristics of Scripting Languages, Uses for Scripting Languages, Web Scripting, and the universe of Scripting Languages. PERL- Names and Values, Variables, Scalar Expressions, Control Structures, arrays, list, hashes, strings, pattern and regular expressions, subroutines.

UNIT II:

Advanced perl Finer points of looping, pack and unpack, file system, eval, data structures, packages, modules, objects, interfacing to the operating system, Creating Internet ware applications, Dirty Hands Internet Programming, security Issues.

UNIT III:

PHP Basics PHP Basics- Features, Embedding PHP Code in your Web pages, Outputting the data to the browser, Data types, Variables, Constants, expressions, string interpolation, control structures, Function, Creating a Function, Function Libraries, Arrays, strings and Regular Expressions.

UNIT IV:

Advanced PHP Programming PHP and Web Forms, Files, PHP Authentication and Methodologies -Hard Coded, File Based, Database Based, IP Based, Login Administration, Uploading Files with PHP, Sending Email using PHP, PHP Encryption Functions, the Mcrypt package, Building Web sites for the World.

TEXT BOOKS:

1. The World of Scripting Languages, David Barron, Wiley Publications.
2. Beginning PHP and MySQL, 3rd Edition, Jason Gilmore, Apress Publications (Dream tech)

REFERENCE BOOKS:

1. Open Source Web Development with LAMP using Linux, Apache, MySQL, Perl and PHP, J.Lee and B.Ware (Addison Wesley) Pearson Education.
2. PHP 6 Fast and Easy Web Development, Julie Meloni and Matt Telles, Cengage Learning Publications.
3. PHP 5.1, I.Bayross and S.Shah, The X Team, SPD.
4. Perl by Example, E.Quigley, Pearson Education.

E-RESOURCES:

https://www.perl.org/books/beginning-perl/
https://nptel.ac.in/
https://www.tutorialspoint.com/php/
https://www.tutorialspoint.com/perl/

COMPUTER GRAPHICS and Multimedia Animation
Type of Course : Professional Elective – 1.2

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites:

C- Programming, Mathematical fundamentals.

Course Objectives:

1.. To understand the two-dimensional graphics and their transformations.
2. To understand the three-dimensional graphics and their transformations.
3. To appreciate Visible Surface detection methods.
4. To gain knowledge about Multimedia Applications.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Design and Apply two-dimensional transformations.
CO2	Design and Apply three-dimensional transformations.
CO3	Apply visible surface detection methods for identifying back-faces.
CO4	Understood Different types of Multimedia fundamentals.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	2	2		2								
CO2	2	2		2								
Co3	1	2		2								
CO4	2	2		2								

UNIT I : 2D Primitives

2D Primitives: Application areas of Computer Graphics, Points and Lines, Line, Circle and Ellipse drawing algorithms, Filled area Primitives- Scan line polygon fill algorithm, Boundary fill algorithm, flood fill algorithm.

UNIT II: 2D Modeling

Two dimensional Geometric transformations: Translation, Rotation, Scaling, Reflection, Shear, Matrix representations in Homogeneous coordinates, Composite transformations, Transformations between coordinate systems, Affine Transformations.

Two dimensional viewing: The Viewiing Pipeline, Viewing coordinate Reference Frame, Window-to-View-Port Transformation, Line, Polygon, Curve and Text are clipping algorithms

UNIT III: 3D Modeling

3D Transformations: Translation, Rotation, Scaling, Reflection and Shear Transformations, Composite Transformations.

3D Viewing: Viewing Pipe line, Viewing coordinate Reference frame, Clipping, Parallel and Perspective projections.

Visible surface identification Methods: Back-Face Detection, Z-Buffer, Depth-Sorting, Area Sub Division and Octree Methods.

UNIT IV: MultiMedia Applications

Multimedia Basics- Multimedia Applications, Multimedia system architecture, Evolving technologies for Multimedia, Defining objects for Multimedia systems, Multimedia data interface standards, Multimedia databases.

TEXT BOOKS:

1. Donald Hearn and Pauline Baker M,-“Computer Graphics” Prentice Hall, New Delhi, 2007 [Unit I-III]
2. Andleigh, P.K and Kiran Thakrar,-Multimedia Systems and DesignI,PHI, 2003[Unit-IV]

REFERENCE BOOKS:

1. Hill F S Jr., “Computer Graphics”, Maxwell Macmillan 1990.

E-RESOURCES:

1. https://www.blender.org/support/tutorials.
2. https://www.slideshare.net/deepakmohapatra102/computer-graphics-30123690
3. https://www.academia.edu/6709255/LECTURE_NOTES_on_Computer_Graphics_and_Multimedia_Table_of_Contentshttps://www.academia.edu/5750589/Computer_Graphics_C_Version_by_Donald_Hearn_and_M_Pauline_Bake

Course Code-Data Warehousing and Data Mining
Type of Course :Professional Elective – 1.3

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites:

--

Course Objectives: The objective of this course is to provide knowledge of techniques and strategies to create and use the data warehouses, to understand, learn different data mining techniques and to understand the applicability of these techniques.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Understand the basic concepts of warehousing and mining
CO2	Derive various interesting patterns and associations in datasets.
CO3	Design and develop classifier models to predict future trends.
CO4	Apply unsupervised learning techniques for a given application.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1				3	1							2
CO2	1	2		3	1							
CO3	2	2		3	2							1
CO4	3	2		3	2							2

UNIT I :

UNIT I: Data Warehouse and Online Analytical Processing: Data Warehouse basic concepts, Data Warehouse Modeling: Data cube and OLAP, Data Warehouse Implementation, Data Generalization by Attribute Oriented Induction. Data Preprocessing: Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization.

UNIT II:

Data Mining Introduction: Introduction, Why Data Mining, kinds of Data that can be mined, Patterns that can be Mined, technologies where it can be used, major issues in data Mining. Mining Frequent Patterns, FP growth algorithms, Associations, and Correlations: Basic Concepts, Frequent Item-set Mining Methods

UNIT III:

Classification: Introduction, Decision tree induction, Bayesian Classification, Rule-Based Classification, Techniques to improve Classification Accuracy, Classification by Backpropagation, Support Vector Machines

UNIT IV:

Cluster Analysis: Introduction, overview of basic clustering methods, Partitioning methods, Hierarchical methods, Density-Based Methods: DBSCAN& OPTICS, introduction to outlier analysis

TEXT BOOKS:

[1].Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, Third Edition, Elsevier, 2012.

REFERENCE BOOKS:

[1].G. K. Gupta ,“Introduction to Data Mining with Case Studies”, Easter Economy Edition, Prentice Hall of India, 2006

[2].A Pang-Ning Tan, Michael Steinbach and Vipin Kumar, “Introduction to DataMining”, Second Edition Pearson Education, 2016

[3].K.P. Soman, ShyamDiwakar and V. Ajay “Insight into Data mining Theory and Practice”, Easter Economy Edition, Prentice Hall of India, 2006

E-RESOURCES:

[1]Data Warehouse Tutorial For Beginners | Data Warehouse Concepts | Data Warehousing | Edureka (2017)https://www.youtube.com/watch?v=J326LIUrZM8&t=4s[2]How Artificial Neural Network (Ann) Algorithm Work | Data Mining | Introduction To Neural Network (2016)https://www.youtube.com/watch?v=fwnaijgpih,

Principles of Programming Languages
Type of Course : Professional Elective – 1.4

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites:

C- Programming, Java basics and Operating System basics

Course Objectives:

1. To understand and describe syntax and semantics of programming languages
2. To understand data, data types, and basic statements
3. To understand call-return architecture and ways of implementing them
4. To understand object-orientation, concurrency, and event handling in programming languages
5. To develop programs in non-procedural programming paradigms

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Analyze the basic concepts of programming language, the general problems and methods related to syntax & semantics.
CO2	Interpret the structured data objects, subprograms and programmer defined data types.
CO3	Outline the sequence control and data control.
CO4	Apply the concepts of storage management using programming languages.
CO5	Implementing the Subprogram call and return.
CO6	Classify various programming languages like procedural,non-procedural,structured and object oriented programming language.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	1	2	1	1			1	1	1	1	
CO2	3	2	1	1	1			1	1	1	1	
CO3	3	1	1	1	1			1	1	1	1	
CO4	3	2	1	1	1			1	1	1	1	
CO5	3	1	1	1	1			1	1	1	1	
CO6	3	1	1	1	1			1	1	1	1	

Syllabus :

UNIT I :

SYNTAX AND SEMANTICS: Evolution of programming languages, describing syntax, context,free grammars, attribute grammars, describing semantics, lexical analysis, parsing, recursive - decent bottom - up parsing

UNIT II:

DATA, DATA TYPES, AND BASIC STATEMENTS: Names, variables, binding, type checking, scope, scope rules, lifetime and garbage collection, primitive data types, strings, array types, associative arrays, record types, union types, pointers and references, Arithmetic expressions, overloaded operators, type conversions, relational and boolean expressions , assignment statements , mixed mode assignments, control structures – selection, iterations, branching, guarded Statements

UNIT III:

SUBPROGRAMS AND IMPLEMENTATIONS: Subprograms, design issues, local referencing, parameter passing, overloaded methods, generic methods, design issues for functions, semantics of call and return, implementing simple subprograms, stack and dynamic local variables, nested subprograms, blocks, dynamic scoping

UNIT IV:

OBJECT- ORIENTATION, CONCURRENCY, AND EVENT HANDLING: Object – orientation, design issues for OOP languages, implementation of object, oriented constructs, concurrency, semaphores,
Monitors, message passing, threads, statement level concurrency, exception handling, event handling
FUNCTIONAL PROGRAMMING LANGUAGES: Introduction to lambda calculus, fundamentals of functional
programming languages, Programming with Scheme, – Programming with ML,
LOGIC PROGRAMMING LANGUAGES: Introduction to logic and logic programming, –
Programming with Prolog, multi - paradigm languages

TEXT BOOKS:

1. Robert W. Sebesta, “Concepts of Programming Languages”, Tenth Edition, Addison Wesley, 2012.
2. Programming Langugaes, Principles & Paradigms, 2ed, Allen B Tucker, Robert E Noonan, TMH

REFERENCE BOOKS:

REFERENCES:
1.R.Kent Dybvig, "The Scheme Programming Language",Fourth Edition,MIT Press,2009,
2.Jeffrey D.Ullman, "Elements of ML programming",Second edition,Prentice Hall,1998
3.Richard A.O'Keefe, "The craft of Prolog",MIT Press,2009
4.W.F.Clocksin and C.S.Mellish,"Programming in Prolog:Using the IOS Standard", Fifth Edition,Springer,2004.

E-RESOURCES:

Scheme : https://www.scheme.com/tspl4/
Prolog : http://learnprolognow.org/
Haskell :http://learnyouahaskell.com/

Course Code- Computer Networks LAB

Lecture – Tutorial-Practical::	0-1-2	**Internal Marks:**	40
Credits:	1.5	**External Marks:**	60

Prerequisites: Knowledge of C Programming, Basic commands of UNIX.

Knowledge of C Programming, Basic commands of UNIX

Course Objectives:

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Should be able to Calculate Data link layer farming methods like bit stuffing and byte stuffing.
CO2	Should be able to Analyze Cyclic redundancy check on different polynomials.
CO3	Should be able to understand Socket Programming Implementation by using TCP and UDP Protocols.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	2	2	-	2	-	-	-	-	-	-	-	-
CO2	-	2	2	-	-	-	-	-	-	-	2	2
CO3	3	-	-	2	2	-	-	-	-	-	2	2

List of Programs

1. Understanding and using of commands like ifconfig, netstat, ping, arp, telnet, ftp, finger, traceroute, whois etc. Usage of elementary socket system calls (socket (), bind(), listen(), accept(),connect(),send(),recv(),sendto(),recvfrom()).
2. Implementation of Connection oriented concurrent service (TCP).
3. Implementation of Connectionless Iterative time service (UDP).
4. Implementation of Select(),of getpeername () system call.
5. Implementation of gesockopt (), setsockopt () system calls.
6. Implementation of remote command execution using socket system calls.
7. Implement the data link layer framing methods such as character stuffing and bit stuffing.
8. Implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.
9. Implement Dijkstra‘s algorithm to compute the Shortest path thru a graph.
10. Implementation of Distance Vector Routing Algorithm.
11. Implementation of SMTP.
12. Implementation of FTP.

Note: Implement programs 2 to 6 in C and 8 to 12 in JAVA.

TEXT BOOKS:

1.Tanenbaum and David J Wetherall, Computer Networks, 5th Edition, Pearson Edu, 2010.
2.Computer Networks: A Top Down Approach, Behrouz A. Forouzan , FirouzMosharraf, McGraw Hill Education.

E-RESOURCES:

- http://www.softpanorama.org/Internals/unix_system_calls.shtml
- https://www.tutorialspoint.com/system-calls-in-unix-and-windows

Course Code-Advanced Data Structures Lab

Lecture – Tutorial-Practical::	0-0-3	**Internal Marks:**	40
Credits:	1.5	**External Marks:**	60

Prerequisites: Data Structures, Programming with C

Advance Data Structures

Course Objectives:

Describe and implement a variety of advanced data structures (hash tables, priority queues, balanced search trees, digital search trees).

Analyze the space and time complexity of the algorithms studied in the course.

Identify different solutions for a given problem; analyze advantages and disadvantages to different solutions.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Able to implement Hashing
CO2	Able to implement skip lists
CO3	Able to implement different balanced trees.
CO4	Able to implement heaps and binomial queues.
CO5	Able to implement various string matching algorithms
CO6	Able to implement different search trees

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	1	2	2	-	2	-	-	-	-	-	-	-
CO2	1	1	1	2	1	-	-	-	-	-	-	-
CO3	1	2	3	2	1	-	-	-	-	-	-	-
CO4	2	3	2	2	1	-	-	-	-	-	-	2
CO5	1	2	-	3	3	-	-	-	-	-	-	-
CO6	-	2	3	3	-	-	-	-	-	-	-	2

List of Experiments

Program To implement functions of Dictionary using static Hashing

Program To implement functions of Dictionary using dynamic Hashing

Program To implement Skip Lists
To perform various operations i.e, insertions and deletions on AVL trees

Program To implement Skip Lists

Program To perform various operations i.e, insertions and deletions on AVL trees

Program To perform various operations i.e., insertions and deletions on 2-3 trees.

Program To implement operations on binary heap.

Program To implement operations on Binomial Queue.

Program To implement pattern matching using Boyer-Moore algorithm.

Program To implement Knuth-Morris-Pratt algorithm for pattern matching.

Program To implement Digital Search Tree

Program To implement PATRICIA

Course Code- Basics of Civil and Mechanical Engineering

Lecture – Tutorial-Practical::	0-2-0	**Internal Marks:**	100
Credits:	1	**External Marks:**	

Prerequisites:

1. Learn about the engineering materials, their types, properties and applications.
2. Study the basic manufacturing processes used in manufacturing of products.
3. Understand the concepts of engineering mechanics like analysis of coplanar concurrent systems.
4. Understand the fundamental principles of thermodynamics.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Understand the concepts of force and friction, direction and its application & able to demonstrate the basic surveying skills
CO2	Identify different building materials and their importance in building construction.
CO3	Differentiate brick masonry, stone masonry and types of flooring & roofing.
CO4	Familiarize with the Engineering materials, their types, properties and applications.
CO5	Familiarize with the basic manufacturing processes used in manufacturing of products.
CO6	Analyze coplanar concurrent systems and Familiarize with fundamental principles of thermodynamics.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	2	-	-	-	-	-	-	2	-	-	-	-
CO2	2	-	-	-	-	-	-	2	-	-	-	-
CO3	2	-	-	-	-	-	-	2	-	-	-	-
CO4	3	1	1									3
CO5	3	2	1									
CO6	3	2	1									

UNIT I :

UNIT – I

SIMPLE STRESS AND STRAINS:

Definition of Mechanics- External and Internal forces-Stress and Strain-Elasticity and Hooke's Law-Relations between elastic constants.

SURVEYING:

Objectives, Types, Principles of Surveying; Measurement of distances and angles

UNIT II:

CIVIL ENGINEERING MATERIALS:

Classification of bricks, Manufacture of bricks, Laboratory and field tests on bricks, stones; Grades of Steel and Cement Concrete.

MASONRY:

Bonds in Brick Masonry, Stone Masonry; Types of Flooring and Roofing.

UNIT III:

Introduction to Engineering materials:

Engineering materials, classification - Properties and applications of Metals: Ferrous and Non-ferrous; Non-metals: Glasses- Ceramics; Polymers: PVC & HDPV;Biomaterials- Composite materials.

Basic Manufacturing Processes: Casting: Classification, Steps involved in making a casting - Advantage of casting and its applications. Welding: Classification of welding processes, Fundamental treatment of various welding processes, Soldering and Brazing.

UNIT-IV

System of forces: Types of Force systems - Coplanar Concurrent Forces - Resultant - Moment of a Force -Resultant of a Force System -Conditions of Equilibrium - Equilibrium analysis of Coplanar Force Systems -Free body diagrams.

Fundamentals of Thermodynamics: System, Control Volume, Surrounding, Boundaries, Universe, Types of Systems, Macroscopic and Microscopic viewpoints, Concept of Continuum, Thermodynamic Equilibrium, Zeroth Law of Thermodynamics - Concept of quality of Temperature. First law of Thermodynamics, First law applied to Non flow systems-simple problems,

TEXTBOOKS:

1. Thermal Engineering, R.K.Rajput, Laxmi publications.
2. A text book of Material science and metallurgy – O.P. Khanna/ Dhanpat rai publications.
3. Engineering Mechanics – Statics and Dynamics - A. K. Tayal, Umesh Publications.
4. Elements of Manufacturing Processes - PARASHAR, B.S. NAGENDRA, MITTAL,
5. Engg. Mechanics - S.Timoshenko & D.H.Young., 4th Edn - , Mc Graw Hill publications.
6. Building Materials, S. S. Bhavikatti, Vices publications House private ltd.
7. Surveying and levelling, R. Subramanian, Oxford University press

REFERENCE BOOKS:

1. Thermodynamics – An Engineering Approach – YunusCengel& Boles /TMH
2. Engineering Mechanics, SS Bhavikatti& KG Rajasekharappa, New Age International
3. Materials Science and Metallurgy – C. Daniel Yesudian, D. G. Harris Samuel.
4. Production Technology, K.L.Narayana, S.V.Ramana& P. Vamsi Krishna, first edition, I.K. Books
5.Engineering Thermodynamics / PK Nag /TMH International, 2006.
6.Production Technology Vol I, O.P. Khanna & M. Lal, Dhanpat Rai Publicati
7.Building Materials, S. K. Duggal, New Age International Publications
8.Fundamentals of surveying, S.K. Roy – PHI learning (P) ltd.

E-RESOURCES:

1. https://nptel.ac.in

COURSE CODE- COMPETITIVE CODING

Lecture – tutorial-practical::	0-0-2*	Internal marks:	100
Credits:	1	External marks:	

Prerequisites: data structures, any programming language (C/C++/JAVA/PYTHON)

Competitive Coding:

The focus of the course is the development and implementation of algorithms, as well as the skills required for programming competitions. The students will learn to select appropriate algorithms for a given problem, integrate multiple algorithms for solving a complex problem, design new algorithms, and implement them in c/c++/java/python. They will also learn skills required for participation in programming contests, which include evaluation of problem difficulty, solving problems in teams, and work under time pressure. Students are expected to complete coding of allotted task by the instructor. The instructor may choose fixed number of programs among the given programs in the unit for each student. The choice of programs for a student in a unit is random. The student is allotted 25 marks after successful execution of allotted programs of each unit.

Course objectives:

1. To impart knowledge of designing solutions to print various pattern or shapes.
2. Be familiar with various number based problems and their solutions.
3. To understand problem concepts based on arrays and design solutions.
4. Be familiar with functions, recursion and to design the implementation of solutions based on recursion.
5. To learn and develop solutions for problems on pointers.
6. To understand and apply solutions based on linked lists, stacks and queues.

Course outcomes:

Upon successful completion of the course, the student will be able to:

Co1	Understand and design solutions for problems based on various printing patterns/shapes
Co2	Understand the basic principles of various number based problems and design solutions
Co3	Apply appropriate algorithm design technique to solving array based application problems
Co4	Identify suitable method to solve problems based on functions and recursion
Co5	Understand and solve problems based on pointers
Co6	Analyze and design solutions based on linked lists, stacks and queues.

Contribution of course outcomes towards achievement of program outcomes (1 – low, 2- medium, 3 – high)

	Po 1	**Po 2**	**Po 3**	**Po 4**	**Po 5**	**Po 6**	**Po 7**	**Po 8**	**Po 9**	**Po 10**	**Po 11**	**Po 12**
Co1	3	3										3
Co2	3	3										3
Co3	3	3										3
Co4	3	3										3
Co5	3	3										3
Co6	3	3										3

Unit I :

Basic problems:

Pattern based problems, triangle pattern, diamond shapes, Pascal shape, Floyd triangle, etc.

Number based problems prime, Armstrong, Strong, Perfect, palindrome, Kaprekar numbers etc.

Code the following

1. Print the multiplication table of a given number up to a another specific number
2. Find m power n
3. Find out generic root of a number by c program
4. Sum of Prime factors of a number
5. Display the factors of a given number
6. Find whether given number is Armstrong or not?
7. Find whether given number is Perfect or not?
8. Find whether given number is Strong or not?
9. Find whether given number is Palindrome or not?
10. Find whether given number is Kaperkar or not?
11. Print following patterns

1.

```
5
5 4
5 4 3
5 4 3 2
5 4 3 2 1
```

2.

```
1
2 3
4 5 6
7 8 9 10
```

3.

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

4.

```
    *
   * * *
  * * * * *
 * * * * * * *
* * * * * * * * *
* * * * * * * * *
 * * * * * * *
  * * * * *
   * * *
    *
```

5.

```
3
44
555
6666
555
44
3
```

6.

```
1
```

```
3*2
6*5*4
10*9*8*7
```

12. Find nth Prime
13. Find first n primes
14. Find 1 to n primes
15. Find whether given number is perfect square or not?

Unit II:

Array based problems (one dimensional and two dimensional) – segregation-making two lists from one list making three lists from one list, list searching techniques, various sorting, special series problems, matrix printing in spiral order, matrix diagonal printing, maze problems, etc.

Character and string based problems – removal of spaces, word reverse in sentence, reverse sentence without reversing words, making string shortest possible palindrome, etc.

Code the following

1. Conversion from uppercase to lower case using c program
2. Counting different characters in a string using c program
3. Code which prints initial of any name
4. Seggregate 0’s& 1’s in a list. seggregate 0’s,1’s & 2’s in an array. Seggregate positive and negative numbers in an array.
5. Find the second last index occurrence of a number in list of elements and if the given number does not occur twice, display -1.
6. Longest sub sequence in an array
7. Find the only number which occurs odd number of times in an array (where all other numbers are even) without counting.
8. Remove duplicates from sorted array
9. Print a given matrix in spiral form Given a 2d array, print it in spiral form. See the following examples.

   Please comment down the code in other languages as well below –

   Input:
   ```
   1   2  3  4
   5   6  7  8
   9  10 11 12
   13 14 15 16
   ```
   Output:
   ```
   1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10
   ```
   Input:
   ```
   1  2  3  4 5  6
   7  8  9 10 11 12
   13 14 15 16 17 18
   ```
   Output:
   ```
   1 2 3 4 5 6 12 18 17 16 15 14 13 7 8 9 10 11
   ```
10. Print the following pattern

Testcase 1:

```
1 1 1 1 1 1 2
3 2 2 2 2 2 2
3 3 3 3 3 3 4
5 4 4 4 4 4 4
```

```
5 5 5 5 5 5 6
7 6 6 6 6 6 6
```

Testcase 2:

```
1 1 1 2
3 2 2 2
3 3 3 4
```

11. Zigzag (or diagonal) traversal of Matrix

Given a 2D matrix, print all elements of the given matrix in diagonal order. For example, consider the following 5 X 4 input matrix.

```
  1    2    3    4
  5    6    7    8
  9   10   11   12
 13   14   15   16
 17   18   19   20
```

Diagonal printing of the above matrix is

```
  1    5    2    9    6    3   13   10    7    4   17   14   11    8   18   15   12   19   16   20
```

12. Check for Matrix Symmetry
13. Removal of spaces in a string
14. Reverse words in sentence. Reverse sentence without reversing words
15. Make a string shortest possible palindrome

Unit III:

Functions and recursion, pointer based problems, function pointers and array pointers

1. Write a c program to print Fibonacci series of given range.
2. Consider the below series : 0,0,2,1,4,2,6,3,8,4,10,5,12,6,14,7,16,8

   This series is a mixture of 2 series all the odd terms in this series form even numbers in ascending order and every even terms is derived from the previous term using the formula (x/2)

   Write a program to find the nth term in this series.

   The value n in a positive integer that should be read from stdin the nth term that is calculated by the program should be written to stdout. You can assume that the n will not exceed 20,000.
3. Consider the following series: 1,1,2,3,4,9,8,27,16,81,32,243,64,729,128,2187…

   This series is a mixture of 2 series – all the odd terms in this series form a geometric series and all the even terms form yet another geometric series. Write a program to find the nth term in the series.

   The value n in a positive integer that should be read from stdin. The nth term that is calculated by the program should be written to stdout. May consider that n not greater than 30.
4. There is a colony of 8 houses represented as cells arranged in a straight line. Each day every cell competes with its adjacent cells (neighbours). Each day, for each cell, if its neighbours are both active or both inactive, the cell becomes inactive the next day,. Otherwise it becomes active the next day.

   Assumptions: the two cells on the ends have single adjacent cell, so the other adjacent cell can be assumed to be always inactive. Even after updating the cell state, consider its pervious state for updating the state of other cells. Update the cell information of all cells simultaneously.

   Write a function cellcompete which takes one 8 element array of integers cells representing the current state of 8 cells and one integer days representing the number of days to simulate. An integer value of 1 represents an active cell and value of 0 represents an inactive cell.
5. The least recently used (lru) cache algorithm exits the element from the cache that was least recently used when the cache is full. After an element is requested from the cache, it should be

added to the cache (if not there) and considered the most recently used element in the cache whether it is newly added or was already existing. Initially, the cache is empty. Implement the function lrucountmiss shall consist of an integer max_cache_size, an array pages and its length len and the function returns an integer indicating the number of cache misses m using the lru cache algorithm execution for the given input. Assume that the array pages always have pages numbered from 0 to 50. (a hit means the requested page is already existing in the cache and a miss means the requested page is not found in the cache). Input format: cache size s, the n pages being requested from the cache and the number of page requests n separated by a space.

6. Print palindrome from num to given number of digits when num is less than the max of given number of digits in the following pattern otherwise print "invalid"

   Input:

   ```
   3   2
       [4,5,6,7,8,9,11,22,33,44,55,66,77,88,99]
   ```

7. Sandwich pattern

   Input:

   5

   Output:

   ```
   1 *2 *3 *4 *5
   11 *12 *13 *14 *15
   21 *22 *23 *24 *25
   16 *17 *18 *19 *20
   6 *7 *8 *9 *10
   ```

   Input: 6

   Output:

   ```
   1 *2 *3 *4 *5 *6
   13 *14 *15 *16 *17 *18
   25 *26 *27 *28 *29 *30
   31 *32 *33 *34 *35 *36
   19 *20 *21 *22 *23 *24
   7 *8 *9 *10 *11 *12
   ```

8. TRAPEZIUM PATTERN

   INPUT:

   4

   OUTPUT:

   ```
   1*2*3*4*17*18*19*20
     5*6*7*14*15*16
       8*9*12*13
         10*11
   ```

9. Longest Increasing Subsequence using Longest Common Subsequence Algorithm
10. Number of sub arrays with negative product Array is {-1,2,-2} No. of negative products are 4.

Unit IV:

Linked lists, Queues, Stack, Graph and tree based problems

1. Merge two sorted lists into third list
2. Implement Queue using Stack
3. Implement Stack using Queue
4. Reverse linked list without using extra space

5. Sort the linked list without using extra space
6. Intersection of two linked lists
7. Swap pair wise nodes in Linked Lists
8. Count of triples (A, B, C) where A*C is greater than B*B, where A, B and C are integers.
9. Design an Algorithm to find articulation point of a tree.
10. Convert a graph to a tree

Text books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, And Clifford Stein, Introduction To Algorithms, First Or Second Edition, Mcgraw Hill.
2. Data Structures And Algorithmic Thinking With Python, By Narasimha Karumanchi, Careermonk Publications
3. Dynamic Programming For Coding Interviews. A Bottom-Up Approach To Problem Solving, By Meenakshi & Kamal Rawat, Notion Press

Reference Books:

1. Cracking The Coding Interview 6th Edition
2. Guide To Competitive Programming: Learning And Improving Algorithms Through Contests (Undergraduate Topics In Computer Science) Springer

E-resources:

www.andrew.cmu.edu/course/15-295,
https://cses.fi/book/book.pdf.
https://www.comp.nus.edu.sg/~stevenha/myteaching/
https://www.javatpoint.com/programs-list
https://practice.geeksforgeeks.org/

SEMINAR

Lecture – tutorial-practical::	0-0-2*	Internal marks:	100
Credits:	1	External marks:	
Prerequisites:			

Learning Outcomes

1. To expose students to a variety of current or emerging topics/research publications and provide them with the opportunity of good public seminars.
2. To attain confidence and comfort on the podium requires practice
3. To learn, practice, and critique effective scientific seminar skills.
4. To develop presentation skills that will be essential during their entire professional careers.
5. To practice within the friendly confines of the department, observed by the faculty and peers

Course outcomes:

Upon successful completion of the course, the student will be able to:

Co1	To enable the students experience and reflect upon their own thinking as it is expressed in communication with others.
Co2	To Examine various newspapers, magazines, articles and books, journals, the web, and other instances of contemporary expression so as to discern genuine thinking from the spurious.
Co3	To learn to work on oral skills like conversational practices, extempore and role play
Co4	To learn the oral presentation techniques(planning preparation practice and presenting)
Co5	To improve presentation skills and develop confidence level in students
Co6	To inculcate the scientific analysis and research on various research topics and get good critical thinking and understanding.

Contribution of course outcomes towards achievement of program outcomes (1 – low, 2- medium, 3 – high)

	Po 1	**Po 2**	**Po 3**	**Po 4**	**Po 5**	**Po 6**	**Po 7**	**Po 8**	**Po 9**	**Po 10**	**Po 11**	**Po 12**
Co1	3	3										3
Co2	3	3										3
Co3	3	3										3
Co4	3	3										3
Co5	3	3										3
Co6	3	3										3

Course Outline

In this course, student will participate in five activities that will hone their oral presentation skills: observation, question, critique, research, and presentation.

Observation: One of the most effective means of gaining an appreciation for the art of presentation is to observe the performance of others. The mannerisms and appearance, the voice, and the visual aids employed by a speaker may be viewed in light of what works and what doesn't. Students learn by example.

Question: The formulation of relevant questions that probe a speaker's knowledge, experimental methods, assumptions, and interpretations is an important part of any presentation, and of the scientific method.

Critique: The critique offers the opportunity for observers to indicate to the presenter (1) areas within the presentation that were well-done, and (2) areas within the presentation that could be improved upon.

Research: It is an expectation and goal of the CSE faculty that this seminar series will be a forum for our students and faculty to dispense and gain insight into the current and cutting edge activities in engineering and technology. To meet these expectations, you will be required to research on current topics/Research publications and develop a presentation.

Presentation: Public speaking is not normally a pleasant experience, particularly for those who are new to the activity; but, it can become so. To attain confidence and comfort on the podium requires practice. This allows students the opportunity to practice within the friendly confines of the department, observed by the faculty and your peers. Students will be responsible for developing and delivering a seminar presentation on a topic that is within one of the discipline areas in the department.

In general, students are expected to create and deliver the following three presentations in the semester.

Series 1 : A power point presentation on any technical topic/text of student's choice. The length of your presentation will be 15 to 20 minutes, followed by a 5 to 10 minutes for questions.

Series 2 : : The learner will be an active and engaged participant by analyzing, constructing/creating, and evaluating information presented in technical and/or scientific journals. The length of the power point presentation will be 15 to 20 minutes, followed by a 5 to 10 minutes for questions.

Series 3 : The learner will conduct a 5 minute video presentation to be delivered based upon a journal article.

Note : 1. Students have to create presentations on their own and downloaded/copied presentations will be severely dealt.

2. Students will have to submit a one page write up to the coordinator at least one week in advance.

3. The write up format will be circulated by the course coordinator.

INDIAN CONSTITUTION
Type of Course : Audit Course

Lecture – Tutorial-Practical::	0-1-2*	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites:

Course Objectives:

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	
CO2	
CO3	
CO4	
CO5	

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	2										2
CO2	3	3	2									
CO3	3		2									2
CO4			3		3							2
CO5	3		3		2							2

UNIT I

1. Meaning of the constitution law and constitutionalism
2. Historical perspective of the Constitution of India
3. Salient features and characteristics of the Constitution of India

UNIT II

4. Scheme of the fundamental rights
5. The scheme of the Fundamental Duties and its legal status
6. The Directive Principles of State Policy – Its importance and implementation

UNIT III

7. Federal structure and distribution of legislative and financial powers between the Union and the States
8. Parliamentary Form of Government in India – The constitution powers and status of the President of India

UNIT IV

9. Amendment of the Constitutional Powers and Procedure
10. The historical perspectives of the constitutional amendments in India
11. Emergency Provisions : National Emergency, President Rule, Financial Emergency

Reference Books

1. **Durgadas Basu – Introduction to the Constitution of India**
2. **Sharma, Sharma B. K. – Introduction to the Constitution of India**
3. **Randhir Sarma Srkar – The Constitution of India**

4. **Govt. of India – The Constitution of India**

III – II Courses

OPERATING SYSTEMS

Lecture – Tutorial-Practical::	2-1-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60
Prerequisites: Basics of Computer Organization.			
Course Objectives:-Students will study and apply concepts related operating systems such as processand memory management, concurrency control, scheduling, I/O and files and mass storage system.			
Course Outcomes:			
Upon successful completion of the course, the student will be able to:			
CO1	describe the important computer system resources and the structure and functioning of operating system, their process management policies and scheduling of processes by CPU.		
CO2	evaluate the requirement for process synchronization and coordination handled by operating system. Describe and analyze the memory management and its allocation policies.		
CO3	understand demand paging, thrashing and principles of deadlocks.		
CO4	understandFile system Interface, File System implementation, Mass-storage structure and disk scheduling algorithms.		

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	2	2	2	1								
CO2	2	3	2	2								
CO3	2	2	2	2								
CO4	2	3	3	2								

UNIT I :

Computer System and Operating System Overview: Overview of computer operating systems, operating systems functions, operating systems structures and systems calls, operating systems generation, **Process Management** – Process concept- process scheduling, operations, Inter process communication. Multi Thread programming models. Process scheduling criteria and algorithms, and their evaluation.

UNIT II:

Concurrency: Process synchronization, the critical- section problem, Peterson’s Solution, synchronization Hardware, semaphores, classic problems of synchronization, monitors, Synchronization examples. **Memory Management**: Swapping, contiguous memory allocation, paging, structure of the page table , segmentation.

UNIT III:

Virtual Memory Management: virtual memory, demand paging, page-Replacement, algorithms, Allocation of Frames, Thrashing. **Principles of deadlock** – system model, deadlock characterization, deadlock prevention, detection and avoidance, recovery form deadlock.

UNIT IV:

File system Interface- the concept of a file, Access Methods, Directory structure, File system mounting, file sharing, protection. **File System implementation**- File system structure, file system implementation, directory implementation, allocation methods, free-space management

Mass-storage structure: overview of Mass-storage structure, Disk structure, disk attachment, disk scheduling, swap-space management.

TEXT BOOKS:

1. Operating System Concepts- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley.
2. Operating Systems – Internal and Design Principles, William Stallings, Sixth Edition–2005, Pearson Education.

REFERENCE BOOKS:

Operating systems- A Concept based Approach-D.M.Dhamdhere, 2nd Edition, TMH.

Operating System A Design Approach-Crowley, TMH.

Modern Operating Systems, Andrew S Tanenbaum 3rd edition PHI.

E-RESOURCES:

https://nptel.ac.in/courses/106105214/

https://www.udacity.com/course/introduction-to-operating-systems--ud923

https://www.youtube.com/watch?v=qf668RboXLs

https://www.youtube.com/watch?v=VoaNyf9iO4Q&list=PLV8vIYTIdSnaHTjrBXjSyNTOWEtA33hvn

Course Code- Design Analysis and Algorithms

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites: Data Structures

Design and Analysis of Algorithms

Course Objectives: Student will

- Analyze the asymptotic performance of algorithms and components
- To study divide and conquer paradigm approach used to analyze and design algorithms
- To study greedy method approach used to analyze and design algorithms.
- To study Dynamic programming   paradigm Backtracking approach used to analyze and design algorithms
- To study Backtracking approach used to analyze and design algorithms
- To study branch and bound paradigm and Deterministic approach used to analyze and design algorithms

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Analyze worst-case running times of algorithms using asymptotic analysis and components
CO2	Describe the divide and conquer method explains when an algorithmic design situation demands it.
CO3	Describe the greedy method explains when an algorithmic design situation demands it.
CO4	Describe the dynamic-programming paradigm explains when an algorithmic design demands it.

CO5	Describe the back tracking method explains when an algorithmic design demands it.
CO6	Describe the branch and bound paradigm and deterministic methods e-plain when an algorithmic design demands it.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	3	3										
CO2	3	3	2									
CO3	3	3	3	2								
CO4	3	3	3	2								
CO5	3	3	3	2								
CO6	3	3	3	1								

UNIT I :

Introduction to Algorithms

Fundamentals of algorithmic problem solving – Analysis framework - Performance Analysis: - Space complexity, Time complexity - Growth of Functions: Asymptotic Notation- Big oh notation, Omega notation, Theta notation, little oh,

Divide and Conquer: Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Finding the Maximum and Minimum

UNIT II:

probabilistic analysis, Amortized analysis

Greedy method: The General Method, Knapsack Problem, Job Sequencing with Deadlines, Minimum-cost Spanning Trees, Prim's Algorithm, Kruskal's Algorithms, Optimal Merge Patterns, Single Source Shortest Paths

UNIT III:

Dynamic Programming: General method, applications-Matrix chain multiplication, Optimal binary search trees, 0/1knapsack problem, All pairs shortest path problem, Travelling sales person problem, Reliability design.

UNIT IV:

Backtracking: General method, applications-n-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

Branch and Bound: General method, applications - Travelling sales person problem, 0/1 knapsack problem- LC Branch and Bound solution, FIFO Branch and Bound solution

P and NP problems, NP-Complete problems

TEXT BOOKS:

Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekaran, University press

REFERENCE BOOKS:

4. Introduction to The Design and Analysis of Algorithms, 3rd Edition, Anany Levitin, Pearson Education, 2017.

5. Introduction to Algorithms, second edition, T.H.Cormen, C.E.Leiserson, R.L. Rivest, and C.Stein, PHI Pvt. Ltd./ Pearson Education
6. Design and Analysis of algorithms, Aho, Ullman and Hopcroft,Pearson education.
7. Algorithms – Richard Johnson Baugh and Marcus Schaefer, Pearson Education.

E-RESOURCES:

Lecture Videos from University of Florida: https://www.cise.ufl.edu/class/cot5405sp18/index
Lecture Videos from NPTEL: https://nptel.ac.in/courses/106106131/

Course Code- <u>COMPILER DESIGN</u>

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites: Formal Language and Automata Theory

Compiler Design

Course Objectives:

1. To describe the design of a compiler including its phases and components and basic understanding of Grammars and language definition.
2. To Identify the similarities and differences among various parsing techniques and grammar transformation Techniques.
3. To Understand the syntax analysis, intermediate code generation, type checking, the role of symbol table and its organization.
4. To Understand, design code generation and optimization schemes.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Describes the major concept and areas of language translation in compilers, the functionality and complexity levels of various translators, linkers, loaders.
CO2	Describes practical experience in phases of compiler.
CO3	Compare and differentiate various parsing and grammar transformation techniques
CO4	Construct intermediate code and performs type checking.
CO5	Schedule symbol table and its organization.
CO6	Illustrate Code generation, obtains machine independent code optimization and instruction scheduling.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	3	3										
CO2	3	3	2									
CO3			3	3								
CO4			3	2								
CO5			3	3								
CO6									3			3

UNIT I :

Overview of Compilation: Overview of language processing – pre-processors – compiler – assembler – interpreters, pre-processors, – linkers &loaders - structure of a compiler – phases of a compiler

Lexical Analysis – Role of Lexical Analysis– Lexical Analysis Vs. Parsing – Token, patterns and Lexemes – Lexical Errors, Recognitions of tokens the lexical analyzer generator lexical

UNIT II:

Top down Parsing: Context free grammars, Top down parsing – Backtracking, First and Follow- LL(1) Grammars, Non-Recursive predictive parsing – Error recovery in predictive parsing.

What is bottom up parsing approach: Introduction to simple LR – Why LR Parsers – Model of an LR Parsers– Difference between LR and LL Parsers,Construction of SLR Tables.

More powerful LR parses, construction of CLR (1), LALR Parsing tables, Dangling ELSE Ambiguity, Error recoveryin LR Parsing. Comparison of all bottoms up approaches with all top down approaches

UNIT III:

Semantic analysis: Semantic analysis, SDT Schemes, evaluation of semantic rules. Intermediate code, three address code, quadruples, triples, abstract syntax trees. Types and declarations, type Checking.

Symbol Tables: Runtime Environments, Stack allocation of space, access to Non Local date on the stack Heap

Management code generation – Issues in design of code generation the target Language Address in the target code Basic blocks and Flow graphs. A Simple Code generation.

UNIT IV:

Machine independent code optimization – semantic preserving transformations, global common sub expression elimination, copy propagation, dead code elimination, constant folding, strength reduction, loop optimization. Instruction scheduling, inter procedural optimization.

Code generation: Object code forms, machine dependent code optimization, register allocation and assignment generic code generation algorithms, DAG for register allocation.

TEXT BOOKS:

1. Alfred V. Aho, Ravi Sethi & Jeffrey. D. Ullman, “Compilers Principles, Techniques & Tools”, Pearson
   Education, third edition, 2007.
2. Andrew N. Appel, “Modern Compiler Implementation in C”, Cambridge University Press, 2004.

REFERENCE BOOKS:

1. John R. Levine, Tony Mason, Doug Brown, “lex & yacc”, O'Reilly Media, Inc., 1992.
2. Kenneth C. Louden, Compiler Construction: Principles and Practice, Course Technology Inc;

International edition, 1997
E-RESOURCES:
1. https://www.holub.com/software/compiler.design.in.c.html

ARTIFICIAL INTELLIGENCE

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60
Prerequisites:			

Course Objectives:

1 To learn the difference between optimal reasoning vs human like reasoning

2 To understand the notions of state space representation, exhaustive search, heuristic search along with the time and space complexities

3 To learn different knowledge representation techniques

4 To understand the applications of AI: namely Game Playing, Theorem Proving, Expert Systems, Machine Learning and Natural Language Processing

Course Outcomes:

Upon Completion of the course, the students will be able to

CO1 Possess the ability to formulate an efficient problem space for a problem expressed in English.

CO 2 Possess the ability to select a search algorithm for a problem and characterize its time and space complexities.

CO3 Possess the skill for representing knowledge using the appropriate technique

CO4 Possess the ability to apply AI techniques to solve problems of Game Playing, Expert Systems, Machine Learning and Natural Language Processing

CO5 Apply the knowledge to develop the solutions for real life problems

CO6 Develop new algorithms to contribute to the research arena

Upon successful completion of the course, the student will be able to:

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)												
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	1	2	1	1			1	1	1	1	
CO2	3	2	1	1	1			1	1	1	1	
CO3	3	1	1	1	1			1	1	1	1	
CO4	3	2	1	1	1			1	1	1	1	
CO5	3	1	1	1	1			1	1	1	1	
CO6	3	1	1	1	1			1	1	1	1	

Unit - I

Introduction, History, Intelligent Systems, Foundations of AI, Sub areas of AI, Applications. Problem Solving – State-Space Search and Control Strategies: Introduction, General Problem Solving, Characteristics of Problem, Exhaustive Searches, Heuristic Search Techniques, Iterative-Deepening A*, Constraint Satisfaction

Unit – II

Logic Concepts and Logic Programming: Introduction, Propositional Calculus, Propositional Logic, Natural Deduction System, Resolution Refutation in Propositional Logic, Predicate Logic, Logic Programming. Representing Knowledge Using Rules: Logic programming, Procedural Vs Declarative knowledge, Forward Vs Backward Reasoning, Matching, Control Knowledge

Knowledge Representation: Introduction, Approaches to Knowledge Representation, Knowledge Representation using Semantic Network, Extended Semantic Networks for KR, Knowledge Representation using Frames, Conceptual dependencies, Scripts

Unit – III

Learning from observation - Inductive learning – Decision trees – Explanation based learning – Statistical Learning methods - Reinforcement Learning

Reasoning under Uncertainty: Introduction to Non-Monotonic Reasoning, Truth Maintenance Systems, Logics for Non-Monotonic Reasoning,

Statistical Reasoning: Bayes Theorem, Certainty Factors and Rule-Based Systems, Bayesian Probabilistic Inference, Bayesian Networks, Dempster- Shafer Theory

Unit IV

Natural Language Processing: Steps in The Natural Language Processing, Syntactic Processing and Augmented Transition Nets, Semantic Analysis, NLP Understanding Systems;

Fuzzy Logic: Crisp Sets, Fuzzy Sets, Fuzzy Logic Control, Fuzzy Inferences & Fuzzy Systems

Planning with state-space search – partial-order planning – planning graphs – planning and acting in the real world

Experts Systems: Overview of an Expert System, Architecture of an Expert Systems, Different Types of Expert Systems, Architectures, Knowledge Acquisition and Validation Techniques, Knowledge System Building Tools, Expert System Shells.

AI Programming languages: Overview of LISP and PROLOG, Production System in Prolog

Text Books

1. Artificial Intelligence, Elaine Rich and Kevin Knight, Tata Mcgraw-Hill Publications
2. Introduction To Artificial Intelligence & Expert Systems, Patterson, PHI publications

References:

1. Artificial Intelligence, George F Luger, Pearson Education Publications
2. Artificial Intelligence : A modern Approach, Russell and Norvig, Printice Hall
3. Artificial Intelligence, Robert Schalkoff, Mcgraw-Hill Publications
4. Artificial Intelligence and Machine Learning, Vinod Chandra S.S., Anand Hareendran S.

Operating Systems & Unix programming Lab

Lecture – Tutorial-Practical::	0-0-2	**Internal Marks:**	40
Credits:	1	**External Marks:**	60

Prerequisites: C-Programming

Operating Systems & Unix programming Lab

Course Objectives:

To understand the design aspects of operating system.
To study the process management concepts & Techniques.
To familiarize students with theLinux environment.
To learn the fundamentals of shell scripting/programming.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Students able to implement CPU scheduling algorithms ,File Organization techniques and paging techniques
CO2	Students able to write shell scripts in Linux platform.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	3	3	2	3	3						
CO2	3	3	3	2	3	3						

EXPERIMENTS :

PART-A

1. Simulate the following CPU scheduling algorithms a) Round Robin b) SJF c) FCFS d) Priority
2. Simulate MVT and MFT
3. Simulate all File Organization Techniques a) Single level directory b) Two level
c) Hierarchical d) DAG
4. Simulate Bankers Algorithm for Dead Lock Avoidance
5. Simulate Bankers Algorithm for Dead Lock Prevention.
6. Simulate all page replacement algorithms a) FIFO b) LRU c) LFU Etc. …
7. Simulate Paging Technique of memory management.
8. Simulate all file allocation strategies a) Sequential b) Indexed c) Linked

PART-B

1. Write a shell script to generate a multiplication table.
2. Write a shell script that copies multiple files to a directory.
3. Write a shell script which counts the number of lines and words present in a given file.
4. Write a shell script which displays the list of all files in the given directory.
5. Write a shell script(small calculator) that adds, subtracts, multiplies and divides the given two integers.
6. Write a shell script to reverse the rows and columns of a matrix.
7. Write a C program that counts the number of blanks in a text file.
8. C program Displaying real time of day for every 60 seconds
9. Write a C program that illustrates the creation of child process using fork system call.
10. Write a C program that illustrates file locking using semaphores.
11. Write a C program that implements a producer-consumer system with two processes.(using semaphores)
12. Write a C program that illustrates the following.
    a)Creating a message queue.
    b)Writing to a message queue.
    c)Reading from a message queue.

TEXT BOOKS:

1. Operating System Concepts- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley.
2. Operating Systems' – Internal and Design Principles Stallings, Sixth Edition–2005, Pearson education
3.Advanced Programming in the UNIX Environment, 3rd EditionW. Richard Stevens, Stephen A. Rago
4.A Practical Guide to Linux Commands, Editors, and Shell ProgrammingMark G. Sobell, Matthew Helmke

REFERENCE BOOKS:

1. Operating systems- A Concept based Approach-D.M.Dhamdhere, 2nd Edition, TMH.
2. Operating System A Design Approach-Crowley, TMH.
3. Modern Operating Systems, Andrew S Tanenbaum 3rd edition PHI.
4. The Linux Programming Interface.A Linux and UNIX System Programming Handbook
Michael Kerrisk.
5. Shell Programming in Unix, Linux and OS XThe Fourth Edition of Unix Shell Programming
Stephen G. Kochan, Patrick Wood
6. Shell ScriptingHow to Automate Command Line Tasks Using Bash Scripting and Shell ProgrammingJaosn Cannon

E-RESOURCES:
https://www.tutorialspoint.com/unix/index.htm https://www.guru99.com/unix-linux-tutorial.html https://www.javatpoint.com/linux-tutorial https://nptel.ac.in/courses/106105214/ https://www.udacity.com/course/introduction-to-operating-systems--ud923 https://www.youtube.com/watch?v=qf668RboXLs

R-PROGRAMMING LAB

Lecture – Tutorial-Practical::	0-0-2	**Internal Marks:**	40
Credits:	1	**External Marks:**	60

Prerequisites:

C- Programming, Mathematical fundamentals.

Course Objectives:

1.Use R for statistical programming, computation, graphics, and modeling,
2. Write functions and use R in an efficient way,
3. Fit some basic types of statistical models
4. Use R in their own research,
5. Be able to expand their knowledge of R on their own.
6. Use Regression models in the visualization of Data Analytics.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	List motivation for learning a programming language.
CO2	Access online resources for R and import new function packages into the R workspace.
CO3	Import, review, manipulate and summarize data-sets in R
CO4	Explore data-sets to create testable hypotheses and identify appropriate statistical tests
CO5	Perform appropriate statistical tests using R Create and edit visualizations

CO6	Ability to analyze different Data Analytics Applications.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2- Medium, 3 – High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	3	3	3	3	2			2	2			2
CO2	3	3	3	3	3			2	2			2
CO3	3	3	3	3	3			2	2			2
CO4	3	3	3	3	2			2	2			2
CO5	3	3	3	3	3			2	2			2
CO6	3	3	3	3	2			2	2			2

1. Implementation of Data Frames and Lists.
2. Implementation of Matirx Addition and Multiplication.
3. Implementation of Quick Sort.
4. Implementation of Binary Search Tree.
5. Implementation of Set Operations.
6. Implementation of Reading and Writing files.
7. Implementation of Graph Operations.
8. Implementation of Corelation.
9. Implementation of ANNOVA.
10. Implementation of Linear Regression.
11. Implementation of Logistic Regression.
12. Implementation of Random Forest.

TEXT BOOKS:

3. The Art of R Programming, Norman Matloff, Cengage Learning
4. R for Everyone, Lander, Pearson

REFERENCE BOOKS:

1. R Cookbook, PaulTeetor, Oreilly.
2. R in Action,Rob Kabacoff, Manning

E-RESOURCES:

1. https://nptel.ac.in
2. https://www.coursera.org/learn/r-programming
3. https://www.r-project.org/

Course Code- Aptitude and Reasoning 2

Lecture – Tutorial-Practical::	0-0-2	**Internal Marks:**	40
Credits:	1	**External Marks:**	60*
Prerequisites:			

Course Objectives:

1. Students will be introduced to various Arithmetic and Reasoning Problems.
2. The students will have acquaintance with various topics like Time, Speed and Distance, Percentages, Data Interpretation etc…
3. Aptitude is designed to assess the logical thinking and how well they are able to think out of the box. These ability tests (All the companies do prefer this test) are strictly timed to assess the speed and accuracy of the students in solving the problems.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Solve the Arithmetic and Reasoning Problems as fast as possible and as simple as possible.
CO2	Exhibits good analytical skills and aptitude skills.
CO3	Perform well in all competitive exams like RRB, SSC, GROUPS, and BANKING

CO4	Clear the aptitude section of exams for higher education like CAT, GMAT, and GRE etc…

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1												
CO2												
CO3												
CO4												
CO5												
CO6												

UNIT I :

1. NUMBER SYSTEM
2. PROBLEMS ON HCF & LCM

UNIT II:

1. DATA SUFFICIENCY
2. ODD ONE OUT

UNIT III:

1. VENN DIAGRAMS
2. ANALOGY

UNIT IV:

1. RATIOS & PROPORTIONS

TEXT BOOKS:

1) **APTIPEDIA, WILEY**
2) **Quantative Aptitude, RS AGARWAL, S.Chand Publishers**

REFERENCE BOOKS:
HOW TO PREPARE FOR Quantative Aptitude, ARUN SHARMA, Mc GRAW HILL

E-RESOURCES:

1. **M4maths**
2. **Crack Aptitude**
3. **Faceprep**

COURSE CODE- HACKATHON

Lecture -Tutorial- Practical::	0-0-2*	Internal marks:	40
Credits:	1	External marks:	60*
Prerequisites: Data Structures, Any Programming Language (C/C++/JAVA/PYTHON), Competitive Coding			

Course Objectives:

1 To build funnel for designing solutions.
2 To harness creativity and expertise of students
3 To spark institute-level hackathons

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Familiar with various problems and their solutions.
CO2	Design innovative solutions for daunting problems
CO3	Crowd source solutions for real time problems
CO4	Improve their analytic and problem solving skills

Contribution of course outcomes towards achievement of program outcomes (1 – Low, 2- Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**

CO1	3	3										3
CO2	3	3										3
CO3	3	3										3
CO4	3	3										3

Course outline:

Hackathon is an initiative to provide students a platform to solve some of the pressing problems and thus inculcate a culture of product innovation and a mindset of problem solving.

The students have the opportunity to work on challenges faced within various situations to create solid solutions. This course is to inculcate students in acquiring employability skills.

The focus of the course is the development and implementation of algorithms, as well as the skills required for programming competitions. The students will learn to select appropriate algorithms for a given problem, integrate multiple algorithms for solving a complex problem, design new algorithms and implement them in any program language of their choice like C/C++/JAVA/PYTHON. They will also learn skills required for participation in programming contests, which include evaluation of problem difficulty, solving problems in teams, and work under time pressure.

Students are expected to complete coding of allotted task by the instructor. The task is allotted to students on the spot. The task is evaluated through online hackathon environments like Hackerrank, Hackerearth, Instacks, etc. The task contains a clear description of the problem including the perquisites, conditions and range of data, etc. The problem description also displays sample test cases and respective outputs. The successful outputs of all sample and hidden test cases in stipulated time is considered as successful completion of Hackathon for the student.

Student need to submit a copy of the Question along with criteria provided, flowchart of their proposed solution, their actual solution and outcomes obtained. These stand a basis for the evaluation.

Such Hackathons are conducted for four sessions.

1. Hackathon based on numericals, patterns, etc.
2. Hackathon based on Strings, files, etc.
3. Hackathon based on web page design for a given problem
4. Hackathon based on database connectivity.

For a single session of Hackathon, the following evaluation process to be followed

The Flowchart of proposed solution for Hackathon is awarded 5 Marks.

The actual successful solution is proportionately awarded a maximum of 10 Marks.

The obtained marks proportionately awarded a maximum of 10 Marks.

Text books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, And Clifford Stein, Introduction To Algorithms, First Or Second Edition, Mcgraw Hill.
2. Data Structures And Algorithmic Thinking With Python, By Narasimha Karumanchi, Careermonk Publications
3. Dynamic Programming For Coding Interviews. A Bottom-Up Approach To Problem Solving, By

Meenakshi & Kamal Rawat, Notion Press
Reference Books: 1. Cracking The Coding Interview 6th Edition 2. Guide To Competitive Programming: Learning And Improving Algorithms Through Contests (Undergraduate Topics In Computer Science) Springer
E-resources:
Hackerrank.com Hackerearth.com Instacks.com

Course Code-BIOLOGY FOR ENGINEERS

Type of Course : Audit course

Lecture – Tutorial-Practical::	2-0-0 (Audit Course)	**Internal Marks:**	40
Credits:	2	**External Marks:**	60*
Prerequisites:			
--			
Course Objectives: The objective of this course is to provide basic knowledge in biology for the engineers and to analyze biological process in computational tools.			
Course Outcomes:			
Upon successful completion of the course, the student will be able to:			
CO1	Describe the fundamental Principles and methods of engineering		

CO2	Identify the functions of different types in bio-molecules
CO3	Describe mechanisms underlying the working of molecular biological processes including enzyme catalysis, metabolic pathways, gene expression.
CO4	Use Excel, MATLAB and other computational tools to quantitatively analyze biological processes.
CO5	
CO6	

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1		3		2								
CO2		3										
CO3		2		3								
CO4		1		2	3							
CO5												
CO6												

UNIT I :

UNIT I:**Introduction and Classification of Living organisms. Introduction:** Fundamental differences between science and engineering by drawing a comparison between eye and camera, Bird flying and aircraft. Biology as an independent scientific discipline. Discuss how biological observations of 18th Century that lead to major discoveries. Examples from Brownian motion and the origin of thermodynamics by referring to the original observation of Robert Brown and Julius Mayor.**Classification:**Classification of living organisms based on (a) Cellularity-Unicellular or multicellular (b) Ultrastructure-prokaryotes or eukaryotes. (c) Energy and Carbon utilization - Autotrophs, heterotrophs, lithotrophs (d) Ammonia excretion –aminotelic, uricotelic, ureotelic (e) Habitat-acquatic, terrestrial (e) Molecular taxonomy-three major kingdoms of life.

UNIT II:

Biomolecules and EnzymesBiomolecules:Biomolecules: Structures of sugars(Glucose and Fructose), starch and cellulose. Nucleotides and DNA/RNA. Amino acids and lipids. Proteins-structure and functions-as enzymes, transporters, receptors and structural elements**Enzymes:** Enzyme classification, Mechanism of enzyme action.Enzymekinetics and kinetic parameters.

UNIT III:

“Genetics is to biology what Newton’s laws are to Physical Sciences” Mendel’s laws, Concept of segregation and independent assortment. Concept of allele. Concepts of recessiveness and dominance. Gene interaction, Epistasis. Meiosis and Mitosis be taught asa part of genetics. Emphasis to be give not to the mechanics of cell division nor the phases but how genetic material passes from parent to offspring.Information Transfer:DNA as a genetic material. Hierarchy of DNA structure-from single stranded to double helix to nucleosomes. Concept of genetic code. Universality and degeneracy of genetic code. Define gene in terms of complementation and recombination.

UNIT IV:

Metabolism :Exothermic and endothermic versus endergonic and exergoinc reactions. Concept of Keq and its relation to standard free energy. ATP as an energy currency. Breakdown of glucose to CO2 + H2O (Glycolysis and Krebs cycle) and synthesis of glucose from CO2 and H2O (Photosynthesis). Energy yielding and energy consuming reactions.Microbiology:Concept of single celled organisms. Concept of species and strains. Identification and classification of microorganisms. Growth kinetics. Ecological aspects of single celled organisms. Microscopy.

TEXT BOOKS:

Reference Books:

[1] Biology:A global approach: Campbell, N. A.; Reece, J. B.; Urry, Lisa; Cain, M, L.; Wasserman, S. A.; Minorsky, P. V.; Jackson, R. B. Pearson Education Ltd

[2] Outlines of Biochemistry, Conn, E.E; Stumpf, P.K; Bruening, G; Doi, R.H., John Wiley and Sons

REFERENCE BOOKS:

[1] Principles of Biochemistry (V Edition), By Nelson, D. L.; and Cox, M. M.W.H. Freeman and Company

[2] Molecular Genetics (Second edition), Stent, G. S.; and Calender, R.W.H. Freeman and company, Distributed by Satish Kumar Jain for CBS Publisher Microbiology, Prescott, L.M J.P. Harley and C.A. Klein 1995. 2nd edition Wm, C. Brown Publishers

E-RESOURCES:

[1].https://bee.cals.cornell.edu/sites/bee.cals.cornell.edu/files/shared/documents/Career_BEE_Final-for-eb.pdf

[2].https://www.teachengineering.org/subjectareas

Course Code- Enterprising and Startup skills
Type of Course : Audit Course

Lecture – Tutorial-Practical::	0-2-0	**Internal Marks:**	40
Credits:	0	**External Marks:**	60

Prerequisites: Creativity, Logical reasoning

Enterprising and Startup skills

Course Objective:

The enable the students develop and systematically apply an entrepreneurial way of thinking that will allow them to identify and create business opportunities that may be commercialized successfully

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	To evaluate the role and importance of entrepreneurship for economic development
CO2	To acquire necessary knowledge and skills required for organising and carrying out entrepreneur activities through training.
CO3	To analyse and apply contemporary project management tools and methodologies
CO4	To learn policies and their support to small and micro enterprises.
CO5	To consider the legal and financial conditions for starting a business venture, evaluate the effectiveness of different entrepreneurial strategies and challenges
CO6	To understand about supportive role of government, financial institutions and educational institutions offering ED Programmes

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2- Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	1		2			2	2		2			
CO2	1		2		2				2			
CO3	1	2			2						3	
CO4	1					2	2					2
CO5	1					2	2	2				2
CO6	1					2	2	2				2

UNIT I :

Entrepreneurship and Training : Importance and growth of **Entrepreneurship** , Characteristics and Qualities of Entrepreneur, Designing Appropriate Training Programmes to inculcate Entrepreneurial Spirit, Feedback and Performance of Trainees. Creativity and Entrepreneurship: Sources and Methods of Ideas Planning.

UNIT II:

Planning and Evaluation of Projects: Growth of the Firm, Project identification and selection, Factors inducing growth , Project Feasibility Study , Post Planning of Project, Project Planning and Control.

UNIT III:

Small and Micro Enterprises: Importance, definitions – policies and their support to MSMEs - growth and growth strategies – sickness in small business and remedies.

UNIT IV:

Institutional Support to Entrepreneur and MSMEs: Role of Government - Role of IDBI, NIESBUD, SISI, DIC, Commercial Banks, Entrepreneurial Development Institutes, Universities and other Educational Institutions offering Entrepreneurial Development Programmes

TEXT BOOKS:

1. Arya Kumar: “Entrepreneurship”, Pearson, Publishing House, New Delhi, 2012.
2. VSP Rao, Kuratko: “Entrepreneurship’, Cengage Learning, New Delhi,
3. K.Ramachandran: “Entrepreneurship Development”, TMH, New Delhi, 2012
4. B.Janakiram, M Rizwana: “Entrepreneurship Development” Excel Books, New Delhi, 2011

REFERENCE BOOKS:

1. Rajeev Roy: “Entrepreneurship”, Oxford University Press, New Delhi,2012
2. P.C.Shejwalkar: “Entrepreneurship Development”, Everest Publishing House, New Delhi, 2011

E-RESOURCES:

1. http://ediindia.ac.in/e-policy/ [Entepreneurial Policy India]
2. http://en.wikipedia.org/wiki/List_of_venture_capital_companies_in_India [Venture Capital]
3. indiavca.org/venture-capital-in-india.html [Venture Capital]

Advanced Data Base Systems

Type of Course: Professional Elective – 2.1

Lecture – Tutorial-Practical:	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites: DBMS,Programming concepts

Course Objectives:

1.Design databases using data models, Query and manage databases .
2. Distinguish between centralized and distributed databases
3.Implement applications involving complex transaction processing.
4.Do query evaluation and query optimization

COURSE OUTCOMES: Upon successful completion of the course, the student will be able to:

1	**Describe** basic database concepts, Data Models, Schemas, Instances, and Components in the DBMS architecture.
2	**Implement** practical solutions to GIS database problems using OO/OR database, spatial database, data warehousing and data mining approaches
3	**Evaluate** simple strategies for executing a distributed query to select the strategy that minimizes the amount of data transfer
4	**Demonstrate** the issues involved in data integration for distributed query processing
5	**Develop** practical skills in the use of these models and approaches to be able to select and apply the appropriate methods for a particular case
6	**Analysed** internal structures, query evaluation and optimization.

UNIT – I

INTRODUCTION

Introduction of object database systems: Structured data types, operations on structured data, encapsulation and ADTS, Inheritance**.**

ORDBMS

Database design for ORDBMS, ORBMS implementation and challenges, OODBMS, comparison of RDBMS, OODBMS and ORDBMS. Introduction to Parallel databases, architectures for parallel databases, Parallel Query Evaluation: Data partitioning and parallelizing sequential operator evaluation code, parallelizing individual operations, and parallel query optimization.

UNIT – II

DISTRIBUTED DATABASES

Introduction to distributed databases: Features of distributed databases vs centralized databases, Why distributed databases.

DDBMS: Levels of transparency, reference architecture for DDB, types of data fragmentation, distribution transparency for read-only and update applications, distributed database access primitives, Integrity constraints in distributed databases

UNIT – III

DISTRIBUTED DATABASE DESIGN

Distributed database design: framework for distributed database design, the design of database fragmentation, allocation of fragments; Distributed Query processing: Equivalence of transformations for queries, transforming global queries into fragment queries, distributed grouping and aggregation functions.

UNIT – IV

QUERY OPTIMIZATION

A framework for query optimization, join queries and general queries. non-join queries in a distributed DBMS, joins in a distributed DBMS, cost based query optimization. DBMS Vs IR systems, Introduction to Information retrieval, Indexing for text search, web search engine, managing text in a DBMS, a data model for XML, Querying XML data, and efficient evaluation of XML queries

TEXT BOOKS:
1. Raghuramakrishnan and Johannes Gehrke, "Database Management Systems", 3rd Edition, TMH, 2006. 2. S Ceri and G Pelagatti, "Distributed databases principles and systems", 1st Edition, TMH, 2008.
REFERENCE BOOKS:
1. Silberschatz, Korth, "Database System Concepts", 6th Edition, TMH, 2010. 2. Elmasri R, Navathe S B, Somayajulu D V L N, and Gupta S K, "Fundamentals of Database Systems", 5th Edition, Pearson Education,2009. 3. C. J. Date, "Introduction to Database Systems", 8th Edition, Pearson Education, 2009.
E-RESOURCES:

Course Code- UML & Design patterns

Type of Course : Professional Elective – 2.2

Lecture – Tutorial-	3-0-0	**Internal Marks:**	40

Practical::			
Credits:	3	**External Marks:**	60

Prerequisites:

Basics of OOPS concepts

Course Objectives:

1. Introducing the Unified Process and showing how UML can be used within the process.
2. Presenting a comparison of the major UML tools for industrial-strength development.
3. introduction to design patterns, practical experience with a selection of central patterns.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Represent the data dependencies of a simple program using UML
CO2	Represent user and programmatic interactions using UML
CO3	Identify the purpose and methods of use of common object-oriented design patterns
CO4	Select and apply these patterns in their own designs for simple programs

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	2	2	2	2	3						2	2
CO2	2	2	2	2	3						2	2
CO3	2	2	2	2	3						2	2
CO4	2	2	2	2	3						2	2

UNIT I :

Introduction to UML: Importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture, Software Development Life Cycle.

Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams. Advanced classes, advanced relationships, Object diagrams : common modeling techniques.

UNIT II:

Behavioral Modeling: Interactions, Interaction diagrams. Use cases, Use case Diagrams, Activity Diagrams.,Events and signals, state machines, state chart diagrams.

Advanced Behavioral Modeling: Architectural Modeling: Components, Deployment, Component diagrams and Deployment diagrams, Common modeling techniques for component and deployment diagrams

UNIT III:

Introduction : What Is a Design Pattern?, Design Patterns in Smalltalk MVC, Describing Design Patterns, The Catalog of Design Patterns, Organizing the Catalog, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.

Creational Patterns : Abstract Factory, Builder, Factory Method, Prototype, Singleton,

UNIT IV:

Structural Patterns: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy.

Behavioral Patterns : Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, Strategy, Template Method, What to Expect from Design Patterns

TEXT BOOKS:

1. The unified Modeling language user guide by Grady Booch, James Rumbaugh , Ivar Jacobson, PEA
2. Design Patterns By Erich Gamma, Pearson Education

REFERENCE BOOKS:

1. Satzinger: Object Oriented Analysis and Design, CENGAGE
2. O'reilly ' s 'Head-First Design Patterns' by Eric Freeman et al, Oreillly
3. 'Applying UML and patterns' by Craig Larman, Pearson

E-RESOURCES:

https://nptel.ac.in/courses/106105153/

Distributed Systems

Type of Course : Professional Elective – 2.3

Lecture – Tutorial – Practical:	3-0-0	**Internal Marks:**	40

Credits:	3	**External Marks:**	60

Prerequisites:

Operating Systems

Course Objectives:

- Provides an introduction to the fundamentals of distributed computer systems, assuming the availability of facilities for data transmission, IPC mechanisms in distributed systems, Remote procedure calls.
- Expose students to current technology used to build architectures to enhance distributed Computing infrastructures with various computing principles.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Describe important characteristics of distributed systems and the salient architectural features of such systems
CO2	Gaining practical experience of inter-process communication and remote invocation in a distributed environment
CO3	Describe the features and techniques used in distributed systems for implementing parallel processing and distributed file systems.
CO4	Describe techniques for implementing mutual exclusion, transaction processing and recovery concepts in distributed environment.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2- Medium, 3 – High)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1									1	2
CO2	2	2	2	1							1	2
CO3	2	2	2	2							2	1
CO4	2	1	2	1							2	3

UNIT I :

Characterization of Distributed Systems: Introduction, Examples of Distributed Systems, Resource Sharing and the Web, Challenges. **System Models:** Introduction, Architectural Models- Software Layers, System Architecture, Variations, Interface and Objects, Design Requirements for Distributed Architectures, Fundamental Models- Interaction Model, Security Model.

UNIT II:

Interprocess Communication: Introduction, The API for the Internet Protocols- The Characteristics of Interprocess communication, Sockets, UDP Datagram Communication, TCP Stream Communication; Client Server Communication;

Distributed Objects and Remote Invocation: Introduction, Communication between Distributed Objects- Object Model, Distributed Object Modal, Design Issues for RMI, Implementation of RMI, Remote Procedure Call, Events and Notifications.

UNIT III:

Operating System Support: Introduction, The Operating System Layer, Protection, Processesand Threads –Address Space, Creation of a New Process, Threads.

Distributed File Systems: Introduction, File Service Architecture; Peer-to-Peer Systems: Introduction, Napster and its Legacy, Peer-to-Peer Middleware, Routing Overlays.

UNIT IV:

Coordination and Alignment: Introduction, Distributed Mutual Exclusion, Elections, Multicast

Communication.

Transactions & Replications: Introduction, System Model and Group Communication, Concurrency Control in Distributed Transactions, Distributed Dead Locks, Transaction Recovery.

TEXT BOOKS:

[1] George Coulouris, Jean Dollimore, Tim Kindberg, “Distributed Systems- Concepts and Design”, Fourth Edition, Pearson Publication

REFERENCE BOOKS:

[1] Ajay D Kshemkalyani, MukeshSighal, “Distributed Computing, Principles, Algorithms and Systems”, Cambridge

[2] Distributed-Systems-Principles-Paradigms-Tanenbaum PHI

E-RESOURCES:

https://nptel.ac.in/courses/106/106/106106168/

https://www.hpcs.cs.tsukuba.ac.jp/~tatebe/lecture/h23/dsys/dsd-tutorial.html

https://courses.cs.ut.ee/MTAT.08.009/2014_fall/uploads/Main/slides15-1.pdf

Adhoc and Sensor networks
Type of Course : Professional Elective – 2.4

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites:

Networks

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

- Understand the concept of mobile ad hoc networks, design and implementation issues and available solutions
- Demonstrate the routing mechanisms and three classes of approaches: proactive, on-demand, and hybrid
- Explain sensor networks and their characteristics. This includes design of MAC layer protocols, understanding of power management, query processing, and sensor databases.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Understand basic concepts of WIRELESS networks and challenges of adhoc and sensor networks
CO2	Classify the design issues and different categories of MAC protocols
CO3	Explain the various adhoc routing protocols and transport layer mechanisms
CO4	Discuss the sensor characteristics and wsn layer protocols and security issues

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**Pso1**	Pso2	Pso3
CO1	3	2											3	1	
CO2	3	3	2	1									3	2	
CO3	3	3	2	1									3	2	
CO4	3	2											3	1	

UNIT I :

Introduction to Ad Hoc Wireless Networks: Characteristics of MANETs, Applications of MANETs, Challenges.

Routing in MANETs: Topology-based versus Position-based approaches, Topology based routing protocols, Position based routing, Other Routing Protocols.

UNIT II:

Data Transmission In MANETs: The Broadcast Storm, Multicasting, Geocasting

TCP over Ad Hoc Networks: TCP Protocol overview, TOP and MANETs, Solutions for TOP over Ad Hoc

UNIT III:

Basics of Wireless Sensors and Applications: The Mica Mote, Sensing and Communication Range, Design issues, Energy consumption, Clustering of Sensors, Applications

Data Retrieval In Sensor Networks: Classification of WSNs, MAC layer, Routing layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs.

UNIT IV:

Security:Distributed Systems Security,Security in Ad Hoc Networks ,Key Management,Secure Routing,Cooperation in MANETs,Wireless Sensor Networks,Intrusion Detection Systems.

TEXT BOOKS:

- Ad Hoc and Sensor Networks — Theory and Applications, Car/osCorderlo Dharma R Aggarwal, World Scientific Publications /Cambridge University Press, March 2006
- Wireless Sensor Networks: An Information Processing Approach, Feng Zhao, Leonidas Guibas, Elsevier Science imprint, Morgan Kauffman Publishers, 2005, rp 2009.

REFERENCE BOOKS:

- Adhoc Wireless Networks — Architectures and Protocols, C.Siva Ram Murthy, B.S.Murthy, Pearson Education, 2004
- Wireless Sensor Networks — Principles and Practice, Fei Hu, Xiaojun Cao, An Auerbach book, CRC Press, Taylor & Francis Group, 2010
- Wireless Ad hoc Mobile Wireless Networks — Principles, Protocols and Applications, Subir Kumar Sarkar, et al., Auerbach Publications, Taylor & Francis Group, 2008

Computer Vision

Type of Course : Professional Elective – 3.1

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites: Image processing

Course Objectives:

Computer Vision focuses on development of algorithms and techniques to analyze and interpret the visible world around us. This requires understanding of the fundamental concepts related to multi-dimensional signal processing, feature extraction, pattern analysis visual geometric modeling, stochastic optimization etc. Knowledge of these concepts is necessary in this field, to explore and contribute to research and further developments in the field of computer vision. Applications range from Biometrics, Medical diagnosis, document processing, mining of visual content, to surveillance, advanced rendering etc.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Describes the fundamental of image formation and depth estimation of the techniques.
CO2	Describes the feature extraction of the filters.
CO3	Describe the classification of the segmentation techniques and analyze the clusters methods
CO4	Analyze the clusters methods of image patterns
CO5	Illustrate motion analysis of spatio temp techniques.
CO6	Illustrate shape from texture color motion and edges.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	3	2									
CO2	3	3	2									
CO3	3	3	3	2								
CO4	3	3	3	3								
CO5	3	3	3	3								
CO6	3	3	3						3			3

UNIT I :

Digital Image Formation and low-level processing

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

Depth Estimation and Multi Camera Views:

Perspective, Binocular Stereopsis: Camera and Epipolar Geometry; Homography, Rectification, DLT, RANSAC, 3-D reconstruction framework; Auto-calibration

UNIT II:

Feature Extraction:

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

UNIT III:

Image Segmentation:

Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection.

Pattern Analysis:

Clustering: K-Means, K-Medoids, Mixture of Gaussians, Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Classifiers: Bayes, KNN, ANN models; Dimensionality Reduction: PCA, LDA, ICA; Non-parametric methods.

UNIT IV:

Motion Analysis:

Background Subtraction and Modeling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

Shape from X:

Light at Surfaces; Phong Model; Reflectance Map; Albedo estimation; Photometric Stereo; Use of Surface Smoothness Constraint; Shape from Texture, color, motion and edges.

TEXT BOOKS:

1 Richard Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag London Limited 2011.

2 Computer Vision: A Modern Approach, D. A. Forsyth, J. Ponce, Pearson Education, 2003.

REFERENCE BOOKS:

1. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004.
2. K. Fukunaga; Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Morgan Kaufmann, 1990.

E-RESOURCES:

1. https://nptel.ac.in/courses/106105216/
2. https://cw.felk.cvut.cz/wiki/courses/ae4m33mpv/start
3. http://cs.brown.edu/courses/cs143/
4. https://cw.fel.cvut.cz/wiki/courses/a4m33gvg/start

Data Analytics

Type of Course :Professional Elective – 3.2

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites:

--

Course Objectives: The objective of this course is to provide knowledge of understanding the data, apply quantitative statistical analysis and data analytics on data, interpretation of results.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	conduct data analytics using appropriate descriptive and quantitative analysis on real-world problems.
CO2	use data analytics skills like variance, ANOVA, regression techniques.
CO3	Understand the application of nearest neighbor classifiers and the effect of validation techniques on different datasets.
CO4	Apply unsupervised learning techniques for a given applications.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2- Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	2	3	2	3	1							2
CO2	3	3	3	3	1							1
CO3	2	3	2	3	2							1
CO4	3	3	2	3	2							2

UNIT I :

UNIT I:Descriptive Statistics: Introduction to the course for Data Analytics, Descriptive Statistics, Probability Distributions. Inferential Statistics: Inferential Statistics through hypothesis tests Permutation & Randomization Test.

UNIT II:

Regression & ANOVA: Regression, ANOVA (Analysis of Variance).Machine Learning: Introduction and Concepts, differentiating algorithmic and model based Frameworks Regression: Ordinary Least Squares, Ridge Regression, Lasso Regression.

UNIT III:

K Nearest Neighbors Regression & Classification.Supervised Learning with Regression and Classification techniques -1: Bias-Variance Dichotomy, Model Validation Approaches, Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Regression and Classification Trees.

UNIT IV:

Unsupervised Learning: Clustering, Associative Rule Mining. Prescriptive analytics: Creating data for analytics through designed experiments, creating data for analytics through Active learning.

TEXT BOOKS:

[1] Data Analytics Made Accessible by Anil K. Maheshwari, 2015

[2] Too Big to Ignore: The Business Case for Big Data by Phil Simon, 2013 by John Wiley & Sons, Inc.

[3] Data Mining and Business Analytics with R, by Johannes Ledolter, Publisher:Wiley (2013).

[4] An Introduction to Statistical Learning with Application in R, by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Springer (2013).

REFERENCE BOOKS:

[1] Hastie, Trevor, et al.; The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

[2] Montgomery, Douglas C., and George C. Runger.; Applied statistics and probability for engineers. John Wiley & Sons, 2010.

E-RESOURCES:

[1]https://nptel.ac.in/content/syllabus_pdf/110106064.pdf

[2] https://swayam.gov.in/nd1_noc20_ma24/preview

Course Code-Software Testing Methodologies
Type of Course : Professional Elective – 3.3

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites:

Proper knowledge on software engineering and their concepts
Enough knowledge on object oriented modeling and techniques
Knowing the different types and levels of software testing process
Good programming skills and debugging skills

Course Objectives:

Describe the principles and procedures for designing test cases
Provide supports to debugging methods
Acts as the reference for software testing techniques and strategies

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Able To Understand Basic Testing Concepts, Testing Techniques And Strategies
CO2	Have Basic Understanding And Knowledge Of Contemporary Issues Like Component And Interface Testing.
CO3	Able To Support In Generating Test Cases And Test Suites
CO4	Have Basic Understanding And Knowledge About Graphs And Matrix Relations, Apply Testing Methods And Tools

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2- Medium, 3 – High)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	1	3	2						3	2		
CO2			3		3				2	2		1
CO3	3		3				2	1				
CO4	2	2	2		3				3	1		2
CO5												
CO6												

UNIT I :

Introduction: Purpose of Testing, Dichotomies, Model for Testing, Consequences of Bugs,Taxonomy of Bugs.
Flow graphs and Path testing: Basics Concepts of Path Testing, Predicates, Path Predicates andAchievable Paths, Path Sensitizing, Path Instrumentation, Application of Path Testing.
Transaction Flow Testing: Transaction Flows, Transaction Flow Testing Techniques

UNIT II:

Dataflow testing: Basics of Dataflow Testing, Strategies in Dataflow Testing, Application ofDataflow Testing.
Domain Testing: Domains and Paths, Nice & Ugly Domains, Domain testing, Domains andInterfaces Testing, Domain and Interface Testing, Domains and Testability.
Paths, Path products and Regular expressions: Path Products & Path Expression, Reduction Procedure, Applications, Regular Expressions & Flow Anomaly Detection.

UNIT III:

Syntax Testing: Why, What and How, A Grammar for formats, Test Case Generation, Implementation and Application and Testability Tips.
Logic Based Testing: Overview, Decision Tables, Path Expressions, KV Charts, and Specifications.

UNIT IV:

State, State Graphs and Transition Testing: State Graphs, Good & Bad State Graphs, State Testing, and Testability Tips.
Graph Matrices and Application:-Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm.
Software Testing Tools: Introduction to Testing, Automated Testing, Concepts of Test Automation, Introduction to list of tools like Win runner, Load Runner, Jmeter.

TEXT BOOKS:

1. Software testing techniques – Boris Beizer, Dreamtech, second edition.
2. Software Testing- Yogesh Singh, Camebridge

REFERENCE BOOKS:

1. The Craft of software testing - Brian Marick, Pearson Education.
2. Software Testing, 3rd edition, P.C. Jorgensen, Aurbach Publications (Dist.by SPD).
3. Software Testing, N.Chauhan, Oxford University Press.
4. Introduction to Software Testing, P.Ammann&J.Offutt, Cambridge Univ.Press.
5. Effective methods of Software Testing, Perry, John Wiley, 2nd Edition, 1999.
6. Software Testing Concepts and Tools, P.NageswaraRao, dreamtech Press
7. Win Runner in simple steps by Hakeem Shittu, 2007Genixpress.
8. Foundations of Software Testing, D.Graham& Others, Cengage Learning

E-RESOURCES:

Websites
1. Software Testing Training Adwww.talentedge.in/Software-Testing
2. Software Test Techniques www.webcrawler.com
3 Winrunner Testing Tool 4 Performance Testing With JMeter 2.9
e books :
1. Software Testing Tools: Covering Win Runner, Silk Test, ... Dr. K.V.K.K. Prasad - 2004 –
2. Software Testing: Testing SoftwareGerald D. Everett, Raymond McLeod, Jr. – 2007
3.Introducing Software Testing Tamres

Course Code-Cloud Computing Application Development
Type of Course : Professional Elective – 3.4

Lecture – Tutorial-Practical::	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites:

1.Basics of Computer Architecture and Organization
2.Basic OS and Networking concepts
3.Data Structures and Algorithms

Course Objectives:

1. Basics of cloud computing.
2. Key concepts of virtualization.
3. Different Cloud Computing services
4. Cloud Implementation, Programming and Mobile cloud computing
5. Key components of Amazon Web Services
6. Cloud Backup and solutions

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Able to understand distributed systems models and cloud platforms, virtualization levels and types.
CO2	To know the design principles, architectures, and enabling technologies of cloud platform, and Assessment of MapReduce, BigTable, Twister, Dryad, DryadLINQ, Hadoop, Sawzall, and Pig Latin
CO3	Use public cloud like IBM Bluemix, Amazon AWS, Google cloud platform or Microsoft Azure for developing an application
CO4	Work with real cloud services

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	2	2	-	-	-	-	-	-	-	-	-
CO2	3	3	2	2	3	-	-	-	-	-	-	-
CO3	3	3	2	2	-	-	-	-	-	-	-	-
CO4	2	3	3	2	-	-	-	-	-	-	-	-

UNIT I :

Systems modeling, Clustering and virtualization:

Scalable Computing over the Internet, Technologies for Network based systems, System models for Distributed and Cloud Computing, Software environments for distributed systems and clouds, Performance, Security And Energy Efficiency

Virtual Machines and Virtualization of Clusters and Data Centers:

Implementation Levels of Virtualization, Virtualization Structures/ Tools and mechanisms, Virtualization of CPU, Memory and I/O Devices, Virtual Clusters and Resource Management, Virtualization for Data Center Automation.

UNIT II:

Cloud Platform Architecture:

Cloud Computing and service Models, Architectural Design of Compute and Storage Clouds, Public Cloud Platforms, Inter Cloud Resource Management, Cloud Security and Trust Management. Service Oriented Architecture, Message Oriented Middleware.

Cloud Programming and Software Environments:

Features of Cloud and Grid Platforms, Parallel & Distributed Programming Paradigms, Programming Support of Google App Engine, Programming on Amazon AWS and Microsoft Azure, Emerging Cloud Software Environments.

UNIT III:

Cloud Based Applications : developing web service, Understanding cloud ecosystem- SaaS/PaaS, Popular APIs

Designing Code for The Cloud: Designing cloud infrastructure; Web Browsers and the Presentation Layer- Understanding Web browsers attributes and differences. Building blocks of the presentation layer: HTML, HTML5, CSS, Silver-light, flash, java script, JQuery, Boot Strap

Web Development Techniques and Frameworks: Working with AJAX controls, JQuery, JSON, XML, REST. Working on Application development Frameworks e.g. Ruby on Rails ,.Net, Java API's or JSF; Deployment Environments – Platform As A Service(PAAS) ,Amazon, vmForce, Google App Engine, Azure, Heroku, AppForce

UNIT IV:

Developing and Deploying an Application in the real cloud : Building on the experience of the first project students will study the design, development, testing and deployment of an application in the cloud using a development framework and deployment platform

Using real cloud services: Working with compute, Data intensive services, load balancing and scaling services available on real cloud platforms

TEXT BOOKS:

1. Distributed and Cloud Computing, Kai Hwang, Geoffry C. Fox, Jack J. Dongarra MK Elsevier.
2. Cloud Computing, Theory and Practice, Dan C Marinescu, MK Elsevier.
3. Chris Hay, Brian Prince, “Azure in Action” Manning Publications [ISBN: 978-1935182481],2010.
4.Eugene Ciurana, “Developing with Google App Engine” Apress; 1 edition[ISBN: 978-1430218319],2009

REFERENCE BOOKS:

1. Cloud Computing, A Practical Approach, Anthony T Velte, Toby J Velte, Robert Elsenpeter, TMH
2. Mastering Cloud Computing, Foundations and Application Programming, Raj Kumar Buyya, Christen vecctiola, S Tammarai selvi, TMH

E-RESOURCES:

1. https://nptel.ac.in/courses/106/105/106105167/
2. https://nptel.ac.in/courses/106104182/
3. https://www.coursera.org/learn/cloud-applications-part1/home/welcome

Open Elective-II

Name of the Course: Web Technology

Lecture – Tutorial-Practical:	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites: Programming Language like C++/Java.

Course Objectives:

To learn the fundamentals of Internet,HTML,CSS.
To get a understanding of Java Scripts, DHTML.
To learn the fundamentals of XML.
To get a strong understanding of PHP.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Students are able to design HTML pages.
CO2	Students are able to use CSS.
CO3	Students are able to develop programs using Java Scripts.
CO4	Students are able to use DHTML.
CO5	Students are able to work with XML.
CO6	Students are able to design and develop PHP programs.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2- Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	2	3	1	3	2	-	-	2	-	3	-
CO2	-	3	2	1	2	1	-	-	2	-	-	-
CO3	2	2	2	2	3	2	-	-	3	-	-	-
CO4	1	2	2	1	3	2	-	-	3	-	-	-
CO5	1	2	3	3	3	2	-	-	-	-	3	-
CO6	2	3	3	2	3	2	-	-	3	-	3	-

UNIT I

Introduction to the Internet: The World Wide Web, Web Browsers, Web Servers, Uniform Resource Locators, Multipurpose Internet Mail Extensions, The Hypertext Transfer Protocol. Common Gateway Interface(CGI).

HTML tags, Lists, Tables, Images, forms, Frames.

CSS: Levels of Style Sheets, Style Specification Formats, Selector Forms.

UNIT II

The Basic of Javascript: Objects, Primitives Operations and Expressions, ScreenOutput and Keyboard Input, Control Statements, Object Creation and Modfication, Arrays, Functions.

DHTML: Positioning Moving and Changing Elements

UNIT III

Working with XML: Document Type Definition, XML schemas, Document object model, XSLT, DOM and SAX.

UNIT IV

PHP Programming: Introducing PHP: Creating PHP script, Running PHP script. Working with variables and constants: Using variables, Using constants, Data types,Operators.Controlling program flow: Conditional statements,Control statements,Arrays,functions.Working with forms and Databases such as MySQL

TEXT BOOKS:

1. Programming the World Wide Web, Robet W Sebesta, 7ed, Pearson.
2. Web Technologies, Uttam K Roy, Oxford Java Server Pages , Hans Bergstan, Oreilly

REFERENCE BOOKS:

1. Web Technologies, HTML< JavaScript, PHP, Java, JSP, XML and AJAX, Black book, Dream Tech
2. An Introduction to Web Design, Programming, Paul S Wang, Sanda S Katila, Cengage
3. An introduction to Web Design and Programming, Wang Thomson
4.Web application technologies concepts, Knuckles, John Wiley.
5.Programming world wide web, Sebesta, Pearson
6.Beginning Web Programming, Jon Duckett ,Wrox, Wiley Java server pages, Pekowsky, Pearson

E-RESOURCES:

1. https://www.w3schools.com/
2. https://www.tutorialspoint.com/perl/
3. https://www.railstutorial.org/book
4. https://www.cs.usfca.edu/~galles/visualization/Algorithms.html

Open Elective-III

Name of the Course: R Programming

Lecture – Tutorial-Practical:	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites: Programming Language like C.

Course Objectives:

- To learn the fundamentals of R programming
- To get a solid understanding of R functions and data structures
- To demonstrate the graphs using R Programming.
- To demonstrate predefined functions like regression, probability distribution etc.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Students are able to build basic software for real needs.
CO2	Students are able to use Data Structures for solving real needs.
CO3	Students are able to use predefined functions.
CO4	Students are able to plot graphs.
CO5	Students are able to use predefined probability distribution functions.
CO6	Students are able to use regression functions.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2- Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	1	2	1	3	1	-	-	3	-	3	-
CO2	3	3	3	2	3	2	-	-	2	-	-	-
CO3	2	3	3	2	3	-	-	-	3	-	-	-
CO4	3	3	3	2	3	2	-	-	3	-	-	-
CO5	2	2	3	3	2	2		-	-	-	3	-
CO6	1	2	2	2	2	2	-	-	3	-	3	-

UNIT I

Introduction, How to run R, R Sessions and Functions, Basic Math, Variables, Data Types, Vectors, Conclusion, Advanced Data Structures, Data Frames, Lists, Matrices, Arrays, Classes.

R Programming Structures, Control Statements, Loops, - Looping Over Nonvector Sets,- If-Else, Arithmetic and Boolean Operators and values, Default Values for Argument, Return Values, Deciding Whether to explicitly call return- Returning Complex Objects, Functions are Objective, No Pointers in R, Recursion.

UNIT II

Doing Math and Simulation in R, Math Function, Cumulative Sums and Products-Minima and Maxima, Calculus, Functions for Statistical Distribution, Sorting, Linear Algebra Operation on Vectors and Matrices, Set Operation, Input /Out put, Accessing the Keyboard and Monitor, Reading and writer Files,

Graphics, Creating Graphs, The Workhorse of R Base Graphics, the plot() Function – Customizing Graphs, Saving Graphs to Files.

UNIT III

Probability Distributions, Normal Distribution- Binomial Distribution- Poisson Distributions Other Distribution, Basic Statistics, Correlation and Covariance, T-Tests,-ANOVA.

UNIT IV

Linear Models, Simple Linear Regression, -Multiple Regression Generalized Linear Models, Logistic Regression, - Poisson Regression- other Generalized Linear Models-Survival Analysis,Nonlinear Models, Splines- Decision- Random Forests,

TEXT BOOKS:

1) The Art of R Programming, Norman Matloff, Cengage Learning
2) R for Everyone, Lander, Pearson

REFERENCE BOOKS:

1) R Cookbook, PaulTeetor, Oreilly.
2) R in Action,Rob Kabacoff, Manning

E-RESOURCES:

1) Geeks for geeks
2) W3schools
3) Tutorialspoint

Open Elective-III

Name of the Course: Python Programming

Lecture – Tutorial-Practical:	3-0-0	**Internal Marks:**	40
Credits:	3	**External Marks:**	60

Prerequisites: Programming Language like C++/Java.

Course Objectives:

- To learn the fundamentals of python programming
- To get a solid understanding of python functions and data structures
- To demonstrate the use of python lists and dictionaries.
- To implement methods and functions to improve readability of programs.
- Students able to describe and apply object-oriented programming methodology.
- Students able to build software for real needs and prior introduction to testing software.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

CO1	Students are able to build basic software for real needs.
CO2	Students are able to use Data Structures for solving real needs.
CO3	Students are able to use functions and modules.
CO4	Students are able to apply object-oriented programming methodology
CO5	Students are able to handle exceptions in the program.
CO6	Students are able to design, code ,test and debug python language programs.

Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2- Medium, 3 – High)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	1	2	1	3	1	-	-	3	-	3	-
CO2	3	3	3	2	3	2	-	-	2	-	-	-
CO3	2	3	3	2	3	-	-	-	3	-	-	-
CO4	3	3	3	2	3	2	-	-	3	-	-	-
CO5	2	2	3	3	2	2		-	-	-	3	-
CO6	1	2	2	2	2	2	-	-	3	-	3	-

UNIT I

Introduction:History of Python, Need of Python Programming, Applications Basics of Python Programming Using the REPL(Shell), Running Python Scripts, Variables, Assignment, Keywords, Input-Output, Indentation.

Types, Operators and Expressions:Types - Integers, Strings, Booleans; Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations

Control Flow- if, if-elif-else, for, while, break, continue, pass

UNIT II

Data StructuresLists - Operations, Slicing, Methods; Tuples, Sets, Dictionaries, Sequences.

Comprehensions.

Functions- Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Anonymous Functions, Fruitful Functions(Function Returning Values), Scope of the Variables in a Function - Global and Local Variables, Modules

Python packages, Introduction to PIP, Installing Packages via PIP, Using Python Packages

UNIT III

Object Oriented Programming OOP in Python:Classes, 'self variable', Methods, ConstructorMethod, Inheritance, Overriding Methods, Data hiding,

Error and Exceptions**:** Difference between an error and Exception, Handling Exception, tryexcept block, Raising Exceptions, User Defined Exceptions

UNIT IV

Why testing is required ?, Basic concepts of testing, Unit testing in Python, Writing Test cases, Running Tests.

TEXT BOOKS:

1. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
2. Learning Python, Mark Lutz, Orielly

REFERENCE BOOKS:

1. Think Python, Allen Downey, Green Tea Press
2. Core Python Programming, W.Chun, Pearson.
3. Introduction to Python, Kenneth A. Lambert, Cengage

E-RESOURCES:

4) Geeks for geeks
5) W3schools
6) Tutorialspoint