VISION OF THE INSTITUTION

To ignite the minds of the students through academic excellence so as to bring about social transformation and prosperity.

MISSION OF THE INSTITUTION

1. To expand the frontiers of knowledge through Quality Education.
2. To provide valued added Research and Development.
3. To embody a spirit of excellence in Teaching, Creativity, Scholarship and Outreach.
4. To provide a platform for synergy of Academy, Industry and Community.
5. To inculcate high standards of Ethical and Professional Behavior.

VISION OF CSE DEPARTMENT

To build a strong teaching-learning base with a flair for innovation and research that responds to the dynamic needs of the software industry and the society.

MISSION OF CSE DEPARTMENT

1. To provide strong foundation both in theory and applications of Computer Science & Engineering, so as to solve real-world problems
2. To empower students with state-of-art knowledge and up to date technological skills, making them globally competent
3. To promote research, innovation and entrepreneurship with focus on industry and social outreach
4. To foster civic minded leadership with ethics and values among students

PROGRAM EDUCATIONAL OBJECTIVES OF CSE DEPARTMENT

1. Graduates will have knowledge of mathematics, science, engineering fundamentals, and in-depth studies in Computer Science Engineering, and will be able to apply them for formulating, analysing and solving real world problems.
2. Graduates will succeed in earning coveted entry level positions in leading Computer Software and Hardware Firms in India and abroad.
3. Graduates will succeed in the pursuit of advanced degrees and research in engineering or other fields and will have skills for continued, independent, lifelong learning and professional development throughout life.
4. Graduates will have good communication skills, leadership qualities, ethical values and will be able to work in teams with due attention to their social responsibilities.

PROGRAM OUTCOMES OF CSE DEPARTMENT

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES OF CSE DEPARTMENT

1) An ability to demonstrate basic knowledge in databases, programming languages and algorithm analysis in the development of software applications.

2) An ability to design and develop projects using open source tools and efficient datastructures.

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course Structure for B. Tech. (With effect from 2019-2020)

III YEAR I SEMESTER							
S.No	**Subjects**	**L**	**T**	**P**	**C**	**I**	**E**
1	Formal Languages & Automata Theory (FLAT)	3	-	-	3	40	60
2	Design and Analysis of Algorithms	3	-	-	3	40	60
3	Operating Systems	3	-	-	3	40	60
4	**Professional Elective I**						
	1. Software Testing Methodologies	3	-	-	3	40	60
	2. Natural Language Processing						
	3. Full Stack Web Development						
	4. Human Computer Interaction						
5	**Open Elective I (Inter Disciplinary Elective I)**						
	1. Electronic Devices and Circuits(EDC)	3	-	-	3	40	60
	2. Robotics						
	3. Embedded Systems						
	4. Integrated Circuits and Applications						
	5. Statistics with R Programming						
6	Mathematics-III (Differential Calculus and Number Theory & Applications)	2	1	-	3	40	60
7	Socially Relevant Projects (15 Hrs /Sem)	-	-	1	0.5	20	30
8	Quantitative Aptitude II	2	-	-	1	20	30
9	PE-I Lab	-	-	3	1.5	40	60
10	Operating System & Language Processor Lab	-	-	3	1.5	40	60
	Total	**19**	**1**	**7**	**22.5**	**360**	**540**
						900	

III YEAR I SEMISTER
COMPUTER SCIENCE AND ENGINEERING
R19 SYLLABUS

Year/Semester	III B. Tech/I Sem	L	T	P	C
Regulation Year	2019-20	3	0	0	3
Subject	Formal Languages and Automata Theory				

COURSE OBJECTIVES:

1. This subject will introduce students to theformal languages and grammars, automata theory, various automata models and computability.
2. To introduce the concepts in automata theory and theory of computation to design grammars and recognizers for different formal languages.
3. To employ finite state machines to solve problems in computing.
4. To introduce finite state machines, context free grammars and Turing Machines and their properties asthe basis for the formal expressivity of computer languages for solving linguistic decision problems.

UNIT I:

Fundamentals of Automata: Computation, Finite state machine, Components of Finite State Automata, Elements of Finite State System, Mathematical representation of Finite State Machine, Automata Classification, Automata in real world.

UNIT II:

Formal Language Theory: Symbols, Alphabets and Strings, Operations on strings, Formal Languages, Operations on Languages, Formal Language/ Grammar Hierarchy: Formal Languages, Regular Languages, Context-Free Language, Context sensitive language, Recursive language, Recursively Enumerable Language, Other forms of formal languages, Relationship between Grammars and Languages.

UNIT III:

Finite Automata: Introduction, Deterministic Finite Automata (DFA), Design of DFA's, Non Deterministic Finite Automata (NFA), Design of NFA's, Non-Deterministic Automata with ∈ moves, Design of NFA with ∈ moves, Advantages of NFA, NFA versus DFA. Equivalent Automata:Equivalent Finite-State Automata, Equivalence of NFA/NFA-∈ and DFA, Equivalence of NFA with ∈ -moves to NFA without ∈ -moves.

UNIT IV:

Regular Expressions and Languages: Regular languages, Regular expressions, Components of Regular Expression, Properties of Regular Expression, Uses of Regular Expression.Finite Automata and Regular Expressions: Properties of regular sets and regular languages, Arden's Theorem, Equivalence of finite automata and Regular expressions, Equivalence of DFA and regular expression, Equivalence of NFA and Regular expression.

UNIT V:

Transducers: Moore machine, mealy machine, Difference between Moore and Mealy Machines, Equivalence of Moore and Mealy Machines.

Context Free Grammars and Context-Free languages: Types of Grammar, Ambiguous and Unambiguous, Noam Chomsky's classification of grammar and finite automata, Relation between regular grammar and finite automata.

Simplification of Context Free Grammar: Simplification of Context-Free Grammars- Elimination of useless symbols, epsilon productions and Unit productions, Normal Forms for Context free grammars, Chomsky Normal form, Greibach Normal Form, Chomsky Vs Greibach Normal Form, and Application of Context-Free Grammars.

UNIT VI:

Push Down Automata: Definition, Model, Graphical Notation, Instantaneous Description, Language Acceptance of pushdown Automata, Design of Pushdown Automata.

Turing Machine: Introduction, Components of Turing Machine, Description of Turing Machine, Elements ofTM, Moves of a TM, Language accepted by a TM, Role of TM's , Design of TM's.

TEXT BOOKS:

1. Introduction to Automata Theory, Languages and Computation, J.E.Hopcroft, R.Motwani and J.D.Ullman, 3rd Edition, Pearson, 2008.
2. A Text Book on Automata Theory, Nasir S.F.B, P.K.Srimani, Cambridge university press.
3. Theory of Computer Science-Automata, Languages and Computation, K.L.P. Mishra and N.Chandrasekharan, 3rd Edition, PHI, 2007.
4. Elements of Theory of computation, Harry R Lewis,Papdimitriou, PHI. T5: Introduction to Theory of Computation – Sipser 2nd edition CENGAGE.

REFERENCE BOOKS:

1. Formal languages and automata theory,C.K.Nagpal, OXFORD.
2. Theory of Computation, a problem solving approach, Kavi Mahesh, Wiley.
3. Automata, computability and complexity, Theory and applications, Elaine rich, PEARSON.Theory of Computation, Vivek kulkani, OXFORD.

COURSE OUTCOMES:

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

CO1: Understand the fundamentals of automata theory.

CO2: Learn the basics of formal languages, various grammars and their hierarchy.

CO3: Understand and model different types of Finite automata.

CO4: Explain about regular expressions and grammars.

CO5: Describe various transducers, Context free languages and context free grammars.

CO6: Explain and model various problems using Push Down Automata and Turing Machines.

Year/Semester	III B. Tech/I Sem	L	T	P	C
Regulation Year	2019-20	3	0	0	3
Subject	Design and Analysis of Algorithms				

COURSE OBJECTIVES:

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

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

UNIT-I:

Introduction: What is an Algorithm, Algorithm Specification, Pseudocode Conventions Recursive Algorithm, Performance Analysis, Space Complexity, Time Complexity, Amortized Complexity, Asymptotic Notation, Practical Complexities, Performance Measurement.

UNIT-II:

Divide and Conquer: General Method, Defective Chessboard, Binary Search, Finding the Maximum and Minimum, Merge Sort, Quick Sort, Performance Measurement, Randomized Sorting Algorithms (Quick Sort).

UNIT-III:

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

UNIT-IV:

Dynamic Programming: All - Pairs Shortest Paths, Single – Source Shortest paths General Weights, String Editing, 0/1 Knapsack, Reliability Design.

UNIT-V:

Backtracking: The General Method, The 8-Queens Problem, Sum of Subsets, Graph Coloring, HamiltonianCycles.

UNIT-VI:

Branch and Bound: The Method, Least cost (LC) Search, Control Abstraction for LC-Search, Bounding, FIFO Branch-and-Bound, LC Branch and Bound, 0/1 Knapsack Problem, LC Branch-and Bound Solution, FIFO Branch-and-Bound Solution, Travelling Salesperson.

TEXT BOOKS:

1. Fundamentals of computer algorithms E. Horowitz S. Sahni, University Press

2. Introduction to AlgorithmsThomas H. Cormen, PHI Learning

REFERENCE BOOKS

1. The Design and Analysis of Computer Algorithms, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman

2. Algorithm Design, Jon Kleinberg, Pearson.

COURSE OUTCOMES:

Students who complete the course will have demonstrated the ability to do the following:

- Argue the correctness of algorithms using inductive proofs and invariants.
- Analyze worst-case running times of algorithms using asymptotic analysis.
- Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and conquer algorithms. Derive and solve recurrences describing the performance of divide and-conquer algorithms.
- Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic programming algorithms, and analyze them.
- Describe the greedy paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize greedy algorithms, and analyze them.

Year/Semester	III B. Tech/I Sem	L	T	P	C
Regulation Year	2019-20	3	0	0	3
Subject	OPERATING SYSTEMS				

COURSE OBJECTIVES:

1. Study the basic concepts and functions of operating systems.
2. Understand the structure and functions of OS.
3. Learn about Processes, Threads and Scheduling algorithms.
4. Understand the principles of concurrency and Deadlocks.
5. Learn various memory management schemes.
6. Study I/O management and File systems.

UNIT I:

Introduction to Operating System Concept: Types of operating systems, operating systems concepts, operating systems services, Introduction to System call, System call types.

UNIT-II:

Process Management – Process concept, The process, Process State Diagram, Process control block, Process Scheduling- Scheduling Queues, Schedulers, Operations on Processes, Inter-process Communication, Threading Issues, Scheduling-Basic Concepts, Scheduling Criteria, Scheduling Algorithms.

UNIT-III:

Concurrency: Process Synchronization, The Critical- Section Problem, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization examples.

UNIT-IV:

Memory Management: Swapping, Contiguous Memory Allocation, Paging, structure of the Page Table, Segmentation

Virtual Memory Management: Virtual Memory, Demand Paging, Page-Replacement Algorithms, Thrashing.

UNIT-V:

Principles of deadlock – System Model, Deadlock Characterization, Deadlock Prevention, Detection andAvoidance, Recovery form Deadlock.

UNIT VI:

File system Interface- the concept of a file, Access Methods, Directory structure, File system mounting, andfile sharing.

File System implementation- File system structure, allocation methods, free-space management Mass-storage structure overview of Mass-storage structure, Disk scheduling.

System Protection: Goals of protection, Principles and Domain of protection.

COURSE OUTCOMES:

- Describe various generations of Operating System and functions of Operating System.
- Design various Scheduling algorithms.
- Apply the principles of concurrency.
- Compare and contrast various memory management schemes.
- Design deadlock, prevention and avoidance algorithms.
- Design and Implement a prototype file system.

TEXT BOOKS:

1. Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin and Greg Gagne 9th Edition, JohnWiley and Sons Inc., 2012.
2. Operating Systems – Internals and Design Principles, William Stallings, 7th Edition, Prentice Hall, 2011.
3. Operating Systems-S Halder, Alex A Aravind Pearson Education Second Edition 2016.

REFERENCES:

1. Modern Operating Systems, Andrew S. Tanenbaum, Second Edition, Addison Wesley, 2001.
2. Operating Systems: A Design-Oriented Approach, Charles Crowley, Tata Mc Graw Hill Education",1996.
3. Operating Systems: A Concept-Based Approach, D M Dhamdhere, Second Edition, Tata Mc Graw-HillEducation, 2007.

Year/Semester	III B. Tech/I Sem	L	T	P	C
Regulation Year	2019-20	3	0	0	3
Subject	Software Testing Methodologies				

COURSE OBJECTIVES:

- Teach fundamentals of various testing methodologies.
- Describe the principles and procedures for designing test cases.
- Provide supports to debugging methods.
- Acts as the reference for software testing techniques and strategies.

UNIT-I:

Introduction: Purpose of Testing, Dichotomies, Model for Testing, Consequences of Bugs, Taxonomy of Bugs.

Flow graphs and Path testing: Introduction to Path Testing, Control Flow Graphs, Path Testing-Paths, Nodes and Links, Fundamental path selection criteria, Loops, Predicates, Path Predicates, Predicate Interpretation, Path predicate Expressions, Predicate Coverage, Path Sensitizing, Path Instrumentation.

UNIT-II:

Transaction Flow Testing: Transaction Flows, Transaction Flow Testing Techniques. Dataflow testing: Basics of Dataflow Testing, Strategies in Dataflow Testing.

UNIT-III:

Domain Testing: Domains and Paths, Nice & Ugly Domains, Domain testing, Domains and Interfaces Testing, Paths, Path products and Regular expressions: Path Products & Path Expression, Reduction Procedure.

UNIT-IV:

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 – V:

State, State Graphs and Transition Testing: State Graphs, Good & Bad State Graphs, Equivalent States, StateTesting.

Graph Matrices and Applications:-Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm.

UNIT -VI:

Software Testing Tools: Introduction to Testing, Automated Testing, Introduction to About Win Runner ,Using Win runner, Mapping the GUI, Recording Test, Working with Test, Enhancing Test, Checkpoints, Test Script Language, Putting it all together, Running and Debugging Tests, Analyzing Results, Batch Tests,Rapid Test Script Wizard.

COURSE OUTCOMES:

- Understand the basic testing procedures.
- Apply tools to resolve the problems in Real time environment.
- Ability to apply the process of testing and various methodologies in testing for testing for developed software.
- Ability to write test cases for given software to test it before delivery to the customer.

TEXT BOOKS:

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

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.Nageswara Rao, dreamtech Press
7. Win Runner in simple steps by Hakeem Shittu, 2007 Genixpress.
8. Foundations of Software Testing, D.Graham & Others, Cengage Learning.

Year/Semester	III B. Tech / I Sem	L	T	P	C
Regulation Year	2019-20	3	0	0	3
Subject	Natural Language Processing				

COURSEE OBJECTIVES:

- Introduces fundamental concepts and techniques of NLP
- Provides in-depth understanding of the computational properties of natural languages and thecommonly used algorithms for processing linguistic information
- Understand and carry out proper experimental methodology for training and evaluating empiricalNLP systems
- Explore Machine Learning Techniques used in NLP
- Examine Deep Neural Architectures for Sequence Processing
- Understands Encoder-Decoder architectures and is able to build Machine Translation Models.

UNIT I

Introduction : Natural Language Processing Definition, Origins, applications, challenges, components of modern NLP – Regular Expressions, Words, Corpora, Tokenization, Text Normalization, Minimum Edit Distance.

Language Models: N -grams, Evaluating Language Models - Perplexity, Generalization and zeros, Smoothing – Laplace, Add-k, Interpolation and Backoff

UNIT II

Naive Bayes: Naive Bayes Classifier - Training the NB Classifier - an example, Optimizing for Sentiment Analysis, NB for other text classification tasks, NB as a Language Model, Evaluation: Precision, Recall, F-measure, Test sets and Cross-validation.

UNIT III

Logistic Regression: Generative vs. Discriminative classifiers - components of machine learning classifier - Classification: the sigmoid, binary sentiment classification with sigmoid, Learning in Logistic Regression - The Cross-Entropy Loss function, Gradient Descent, SGD, Mini-batch, Regularization - Multinomial Logistic regression.

UNIT IV

Vector Semantics and Embeddings : Lexical Semantics, Vector Semantics, Words and Vectors, Cosine for measuring similarity, TF-IDF: Weighing terms in the vector, Pointwise Mutual Information (PMI), Applications of the tf-idf or PPMI vector models, Word2vec, Visualizing Embeddings, Semantic properties of embeddings, Bias and Embeddings.

UNIT V

Neural Networks and Neural Language Models : Units, the XOR problem, Feedforward Neural Networks, Feedforward networks for NLP : classification, Feedforward Neural Language Modelling, Training Neural Nets, Training the Neural Language the model.

Sequence Labelling for Parts of Speech and Named Entities : English word classes, Part-of-speech tagging, Named Entities and Named Entity Tagging.

UNIT VI

Deep Learning Architectures for Sequence Processing : Language Models, Recurrent Neural Networks,Managing Context in RNNs: LSTMs and GRUs, Potential Harms from Language Models Machine Translation and Encoder-Decoder Models : The Encoder-Decoder Model, Encoder-Decoder withRNNs

TEXT BOOKS:

1. Speech and Natural Language Processing - Daniel Jurafsky & James H Martin, Pearson Publications

REFERENCE BOOKS:

1. Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, Oreilly Publishers - by Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, HarshitSurana
2. Natural Language Processing with Python: Analysing Text with the Natural Language Toolkit.Oreilly Publishers - Stevem Bord. Ewam Klein, Edward Loper

Year/Semester	III B. Tech / I Sem	L	T	P	C
Regulation Year	2019-20	3	0	0	3
Subject	Full Stack Web Development				

UNIT-I:

Eclipse IDE Overview, Git, Maven, JDBC: JDBC Introduction, JDBC Architecture, Database Overview, JDBC Basics, Driver Manager, Result Set, Connection, Statement, Prepared Statement, Callable Statement, DB Connectivity Steps, Store Image in SQL, Read Image in SQL, JDBC CRUD Application

UNIT-II:

Servlet Basics, Need of Server-Side Programming, Servlet Life Cycle, Services doGet(), doPost(), Destroy(),ServletHelloWorldApplication, Web.xmlStructure, Servlet Directives- include(), forward(), sendRedirictive(), HttpServletRequest, HttpServletResponse, Servlet and JDBC Integration, Servlet, HTML 5, MySQL-JDBC.

JSP: JSP Basics, Creating dynamic Web content with JSP, Scriplet, Declaration.

UNIT-III:

Introduction to HTTP, Parameters, Messages, Request. Response, Methods, Status Code, Header Fields. Registration, Authentication, Caching, URL Encoding. Security

UNIT-IV:

Bootstrap: Bootstrap Introduction, Bootstrap Example, Bootstrap Container, Jumbotron, Buttons, Grid, Pagination, Images.

UNIT-V:

Introduction to PERL, Operators and if statements, Program design and control structures, Arrays, Hashs and File handling, Regular expressions, Subroutines, Retrieving documents from the web with Perl.

UNIT-VI:

Introduction to NodeJS: Modules, HTTP Module, File System, URL Module, NPM, Events, Upload Files, Send an Email, Database access with NodeJS

COURSE OUTCOMES:

- Learn to access Database through Java programs, using Java Database Connectivity (JDBC).
- Create dynamic web pages, using Servlets and JSP.
- Learn about the Hyper Text Transfer Protocol's characteristics
- Design and Build responsive designs with Bootstrap
- Understand the fundamental concepts of Perl
- Install Node.js and execute scripts

TEXT BOOKS:

1. JDBC 4.2, Servlet 3.1, and JSP 2.3 Includes JSF 2.2 and Design Patterns, Black Book, 2ed by K.Santhosh Kumar.
2. Full Stack Java Development with Spring MVC, Hibernate, jQuery, and Bootstrap by Myur ramgir, wiley.
3. Programming Perl, 4ed, Tom Christiansen, Jonathan Orwant, Oreilly (2012)

REFERENCES:

https://www.w3schools.com/bootstrap/default.asp
https://www.tutorialspoint.com/perl/index.htm
https://www.w3schools.com/nodejs/default.asp

Year/Semester	III B. Tech / I Sem	L	T	P	C
Regulation Year	2019-20	3	0	0	3
Subject	Human Computer Interaction				

COURSE OBJECTIVES:

The main objective is to get students to think constructively and analytically about how to design and evaluate interactive technologies.

COURSE OUTCOMES:

1. Explain the capabilities of both humans and computers from the viewpoint of human information processing.
2. Describe typical human–computer interaction (HCI) models, styles, and various historic HCI paradigms.
3. Apply an interactive design process and universal design principles to designing HCI systems.
4. Describe and use HCI design principles, standards and guidelines.
5. Analyze and identify user models, user support, socio-organizational issues, and stakeholder requirements of HCI systems.
6. Discuss tasks and dialogs of relevant HCI systems based on task analysis and dialog design.

UNIT I:

Introduction: Usability of Interactive Systems- introduction, usability goals and measures, usability motivations, universal usability, goals for our profession Managing Design Processes: Introduction, Organizational design to support usability, Four pillars of design, development methodologies, Ethnographic observation, Participatory design, Scenario Development, Social impact statement for early design review, legal issues, Usability Testing and Laboratories

UNIT II:

Menu Selection, Form Fill-In and Dialog Boxes: Introduction, Task- Related Menu Organization, Single menus, Combinations of Multiple Menus, Content Organization, Fast Movement Through Menus, Data entry with Menus: Form Fill-in, dialog Boxes, and alternatives, Audio Menus and menus for Small Displays

UNIT III:

Command and Natural Languages: Introduction, Command organization Functionality, Strategies and Structure, Naming and Abbreviations, Natural Language in Computing Interaction Devices: Introduction, Keyboards and Keypads, Pointing Devices, Speech and Auditory Interfaces, Displays- Small and large.

UNIT IV:

Quality of Service: Introduction, Models of Response-Time impacts, Expectations and attitudes, User Productivity, Variability in Response Time, Frustrating Experiences Balancing Function and Fashion: Introduction, Error Messages, Nonanthropomorphic Design, Display Design, Web Page Design, Window Design, Color.

UNIT V:

User Documentation and Online Help: Introduction, Online Vs Paper Documentation, Reading from paper Vs from Displays, Shaping the content of the Documentation, Accessing the Documentation, Online tutorials and animated documentation, Online communities for User Assistance, The Development Process.

UNIT VI:

Information Search: Introduction, Searching in Textual Documents and Database Querying, Multimedia Document Searches, Advanced Filtering and Searching Interfaces Information Visualization: Introduction, Data Type by Task Taxonomy, Challenges for Information Visualization.

TEXT BOOKS:

- Designing the User Interface, Strategies for Effective Human Computer Interaction, 5ed, Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven M Jacobs, Pearson
- The Essential guide to user interface design,2/e, Wilbert O Galitz, Wiley DreamaTech.

REFERENCE BOOKS:

- Human Computer, Interaction Dan R.Olsan, Cengage ,2010.
- Designing the user interface. 4/e, Ben Shneidermann , PEA.
- User Interface Design, Soren Lauesen , PEA.
- Interaction Design PRECE, ROGERS, SHARPS, Wiley

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course Structure for B. Tech. (With effect from 2019-2020)

Year/Semester	III B. Tech / I Sem	L	T	P	C
Regulation Year	2019-20	3	0	0	3
Subject	Statistics with R Programming				

COURSE OBJECTIVES:

- Use R for statistical programming, computation, graphics, and modeling.
- Write functions and use R in an efficient way,
- Fit some basic types of statistical models and able to analyze data for the purpose of exploration using Descriptive and Inferential Statistics.
- Able to understand Probability and Sampling Distributions
- Learn the creative application of Linear Regression in multivariate context for predictive purposes.
- Be able to expand their knowledge of R on their own in research.

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.

UNIT-II

R Programming Structures - Control Statements, Loops, - Looping Over Non vector 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, A Quicksort Implementation-Extended Example: A Binary Search Tree.

UNIT-III

Doing Math and Simulation in R - Math Function, Extended Example Calculating Probability Cumulative Sums and Products-Minima and Maxima-Calculus, Functions for Statistical Distribution, Sorting, Linear Algebra Operation on Vectors and Matrices, Extended Example: Vector cross Product-Extended, Set Operation, Input/output, Accessing the Keyboard and Monitor, Reading and writer Files.

UNIT-IV

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

UNIT-V

Probability Distributions - Normal Distribution- Binomial Distribution- Poisson Distributions, Basic Statistics, Correlation and Covariance.

UNIT-VI

Linear Models - Simple Linear Regression, Multiple Regression Generalized Linear Models, Logistic Regression, Poisson Regression- other Generalized Linear Models-Survival Analysis.

TEXTBOOKS:

1) The Art of R Programming, A K Verma, Cengage Learning.
2) R for Everyone, Lander, Pearson
3) The Art of R Programming, Norman Matloff, No starch Press.

REFERENCE BOOKS:

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

COURSE OUTCOMES:

Upon successful completion of this course, students should be able to:

- Access online resources for R and import new function packages into the R workspace
- Import, review, manipulate and summarize datasets in R
- Explore datasets to create testable hypotheses and identify appropriate statistical tests.
- Create and edit visualizations in R.
- Define, Calculate, Implement Probability and Probability Distributions to solve a wide variety of problems and perform appropriate statistical tests using R
- Understand, Analyze, Interpret Correlation and Regression to analyze the underlying relationships between different variables.

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course Structure for B. Tech. (With effect from 2019-2020)

Year/Semester	III B. Tech / I Sem	L	T	P	C
Regulation Year	2019-20	2	1	0	3
Subject	Mathematics III (Differential Calculus, Number Theory & Applications)				

COURES OBJECTIVES:

- Make the students learn modeling various physical phenomena as first and higher order PDE andapplications
- Learn the Fourier series of periodic functions and expand a function in sine and cosine series
- Understand the number theory concepts

UNIT-I: First Order Partial Differential Equations

Formation of Partial differential equations by elimination of arbitrary constants and arbitrary functions– solutions of first order linear (Lagrange) equations and nonlinear equations-standard types

UNIT- II: Higher Order Partial Differential Equations

Solutions of Linear Partial differential equations with constant coefficients. RHS terms of the type $e^{ax + by}$, Sin(ax+by), Cos (ax+by), $x^m y^n$. Classification of second order partial differential equations-parabolic, elliptical and hyperbolic.

UNIT - III: Fourier Series

Introduction, Periodic function, Dirichlet's conditions, Fourier series of periodic function, Fourier series at the point of discontinuity, Fourier series of even and odd functions, Half-range Fourier Sine and Cosine series. Fourier series in an arbitrary interval.

UNIT-IV: Applications of Partial Differential Equations

Method of Separation of Variables, Applications to wave equation, heat conduction equation in one dimension

UNIT-V:Basic Number Theory

The Well- ordering principle, properties of integers, division algorithm, Greast Common Divisor(GCD), Relatively prime, Prime numbers, Euclids Lemma, Prime Factorization, Number of Divisors, Euler's Function.

UNIT-VI: Theory of Congruences

Congruences -Properties, Residues modulo, Fermat's Theorem, Wilson's Theorem - Applications

COURSE OUTCOMES:

After undergoing this course, students will be able to:

1. model first order linear and non-linear partial differential equations and solve analytically
2. model higher order partial differential equations and solve analytically
3. compute Fourier series of periodic functions
4. physical problems of engineering like steady and unsteady heat conduction, vibration of string, etc.,
5. find GCD, Prime factorization of Integers
6. solve linear congruences in number theory

TEXT BOOKS:

- B. S. Grewal, Higher Engineering Mathematics, 42nd Ed., Khanna Publishers, New Delhi, 2012
- Elementary Number Theory and lts Applications , Kenneth H. Rosen, ADDISON-WESLEY PUBLISHING COMPANY

REFERENCES:

- T.K.V.Iyengar, B. Krishna Ghandhi, S. Ranganatham and M.V.S.S.N.Prasad, Engineering Mathematics, Volume-I, 12th Ed., S. Chand Publishers, 2014
- B. V. Ramana, Higher Engineering Mathematics, 4th Ed., Tata McGraw Hill, New Delhi, 2009
- Elementary Number Theory , David M Burton, 7th Edition, McGraw Hill.

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course Structure for B. Tech. (With effect from 2019-2020)

Year/Semester	III B. Tech / I Sem	L	T	P	C
Regulation Year	2019-20	2	-	-	1
Subject	Quantitative Aptitude II				

COURSE OBJECTIVES:

Enable the students to

- Know the concepts of partnership and their profit sharing at the end.
- Understand the concept of sets and relation between sets and Venn diagrams.
- Apply the concepts of measures of central tendency and dispersion.
- Know the concepts of Permutations & Combinations and their application in probability.
- Calculate ages of persons in a family using the given data.
- Understand the given data and interpret the required values.

UNIT –I: Business & Partnership

Partnership in business- Working and Sleeping Partners -Division of Shares - Partnership Involved Time and Work problems.

UNIT- II:Set Theory &Venn Diagrams

Basic Concepts of Sets-Operations on Sets – Venn Diagrams- Problems.

UNIT –III: Statistics:

Basics of Statistics -Range -Mean- Median-Mode -Standard Deviation-Problems.

UNIT –IV: Permutations & Combinations and Probability

Basic concepts of Permutations & Combinations - Selection with and without repetition- Circular Arrangements.

Concepts of Probability- Various Events of Probability- Related Problems.

UNIT –V: Ages

Ratio Based - Proportion Based - Equation Based – Average Based - Age Problems.

UNIT – VI : Data Interpretation

Line & Bar Graphs- Pie Charts/Graphs-Table –Based Problems.

COURSE OUTCOMES:

After completing this course, the students will be able to

1. Calculate the profit or remuneration received at the end using the ratio of investments or workmen ship.
2. Evaluate number of persons/objects belonging to a specified category using the concept of Venn diagram.
3. Measure the range, mean, median and mode of the given data, identify the extent of dispersion and interpret the data.
4. Compute various ways of selection or arrangement of persons /objects and predict the probability of doing so.
5. Deduce the ratios/ equations corresponding to ages of persons of a family and calculate the corresponding ages.
6. Analyze the given chart / table and interpret the results from the given data.

TEXT BOOKS:

- Dr. R.S.Aggarwal ,Quantitative Aptitude for competitive Examinations,Sultan Chand Publications, 2017.

REFERENCES:

- Arun Sharma, How to Prepare for Quantitative Aptitude for the CAT, Tata McGrawHillPublishing Company, 2016.
- Dinesh Khattar,The Pearson Guide to Quantitative Aptitude for Competitive Examinations, Pearson India, 2016.

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course Structure for B. Tech. (With effect from 2019-2020)

Year/Semester	III B. Tech / I Sem	L	T	P	C
Regulation Year	2019-20	0	0	3	1.5
Subject	Natural Language Processing Lab				

Experiments

1. Solve the following by writing Regular Expressions in Python
    a. Replace all occurrences of **5** with '**five**' for the given string
    b. For the given list, filter all elements that do *not* contain '**e'**.
    c. For the given input string, display all lines not containing '**start'** irrespective of case.
    d. For the given input list, filter all elements that contains **42** surrounded by word characters.
    e. Validate the CVV number(It should have 3 or 4 digits, It should have a digit between 0-9, Itshould not have any alphabets and special characters)
    f. For the given input string, change whole word **mall** to **1234** only if it is at the start of a line
    g. Check whether the given email address is valid or not.
    h. Check whether the Aadhar number is valid or not (It should have 12 digits, It should not startwith 0 and 1, It should not contain any alphabet and special characters, It should have white space after every 4 digits)
2. Write code snippets to
    a. Tokenize words and sentences.
    b. Perform stemming on the tokens present in the given sentence.
    c. Perform Lemmatization on the tokens present in the given sentence.
3. Write a program to implement the Minimum Edit Distance algorithm.
4. Design a function with the name ngram_converter() that takes in a **sentence** and '**n**' as an argument and converts it into N-grams.
5. Write a program to compute unsmoothed unigrams and bigrams.
6. Build a basic N-gram language model using trigrams of Reuters corpus. Reuters corpus is a collection of 10,788 news documents totaling 1.3 million words.
7. Run N-gram program on two different small corpora of your choice (you might use email text or newsgroups). Now compare the statistics of the two corpora. What are the differences in the most common unigrams between the two? How about interesting differences in bigrams?
8. Add an option to the above program to compute the perplexity of a test set.
9. Implement and Evaluate Naïve Bayes Model for Email Spam filtering task.
10. Implement and Evaluate Naïve Bayes Model for Sentiment Analysis task.
11. Write Python functions to calculate sigmoid, softmax, cross-entropy loss.
12. Create a sample value of Z (weighted sum as in logistic regression) and create the cross-entropy loss function plot showing plots for cost function output vs hypothesis function output (probability value).
13. Train a Text Classifier for E-mail spam detection using Logistic Regression.
14. Train a Text Classifier for Sentiment Analysis using Logistic Regression
15. Design a Sequence labelling task "Part-of-Speech tagging" using Hidden Markov Model.
16. Build a custom Named-Entity-Recognition model using any library (NLTK or spacy)

Year/Semester	III B. Tech / I Sem	L	T	P	C
Regulation Year	2019-22	0	0	3	1.5
Subject	Operating System & Language ProcessorLab				

EXPERIMENTS:

PART-1

1. Simulate the following CPU Scheduling Algorithms

   (a) FCFS (b) SJF (c) Priority (d) Round-Robin

2. Simulate the following

   (a) Multiprogramming with a Fixed number of Tasks (MFT)

   (b) Multiprogramming with a Variable number of Tasks (MVT)

3. Simulate Bankers Algorithm for Dead Lock Avoidance.

4. Write a program to implement the producer-consumer problem using semaphores.

5. Write a program to implement IPC using shared memory.

6. Simulate the following Page Replacement Algorithms

   (a) FIFO (b) LRU (c) LFU

7. Write a program to simulate the following contiguous memory allocation techniques

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

8. Write a program to implement Paging technique for memory management.

9. Simulate the following File Allocation Strategies

   (a) Sequenced (b) Indexed (c) Linked

10. Simulate the following Disk Scheduling Algorithms

   (a) FCFS (b) SSTF (c) SCAN

PART-2

1. Implementation of Symbol Table
2. Develop a lexical analyzer to recognize a few patterns in C (Ex. identifiers, constants, comments,operators etc.)
3. Implementation of Lexical Analyzer using Lex Tool
4. Construction of DAG

Additional Experiments:

1. Write a Program to implement Dining Philosophers problem using Monitors.
2. Simulate given File Organization Techniques: (a) Hierarchical (b) DAG
3. Lex Program to convert abc to ABC.
4. Implementation of Recursive Descent Parser