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

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

Vision of IT Department

To be leaders in Information Technology through excellence in education, research and community outreach.

Mission of IT Department

- To provide quality education in the core principles of Information Technology.
- To enable the students to apply the core concepts to solve real world problems.
- To amplify their potential through research and continuous learning for high quality career.
- To mould them as professionals with ethics and morals.

Program Educational Objectives(PEOs)

PEO1: To provide students with a strong foundation in the mathematical, scientific and engineering fundamentals necessary to formulate, solve and analyze engineering problems.

PEO2: Graduates will succeed in entry-level engineering positions in IT industry and with government agencies.

PEO3: Graduates will succeed in the pursuit of advanced degrees in engineering or other fields and will have skills for, continued independent, lifelong learning to become experts in their profession.

PEO4: Empower students with effective teamwork, communication skills, leadership skills, ethical values and high integrity to serve the interests of the society and nation.

Program Outcomes(POs) of IT 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 (PSOs) of IT Department

1. An ability to demonstrate basic knowledge in databases, programming languages, common business functions and algorithm analysis to design and develop appropriate Information Technology solutions.
2. Ability to organize an IT Infrastructure, manage and monitor resources and secure the data.

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

II YEAR I SEMESTER							
S.No	**Subjects**	**L**	**T**	**P**	**C**	**I**	**E**
1	Software Engineering	3	-	-	3	40	60
2	Discrete Mathematical Structures	2	1	-	3	40	60
3	AI Tools, Techniques & Applications	3	-	-	3	40	60
4	Data Structures & Algorithms	3	-	-	3	40	60
5	Digital Logic Design	3	-	-	3	40	60
6	Object Oriented Programming through Java	3	-	-	3	40	60
7	Quantitative Aptitude I	3	-	-	0	0	0
8	AI Tools, Techniques & Applications Lab	-	-	3	1.5	40	60
9	Data Structures & Algorithms Lab	-	-	3	1.5	40	60
10	Object Oriented Programming through Java Lab	-	-	3	1.5	40	60
	Total	**20**	**1**	**9**	**22.5**	**360**	**540**
						900	

II YEAR I SEMISTER
INFORMATION TECHNOLOGY
R19 SYLLABUS

Year/Semester	II B. Tech/I Sem	L	T	P	C
Regulation Year	2020-21	3	0	0	3
Subject	Software Engineering				

COURSE OBJECTIVES:

1. To help students to develop skills that will enable them to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain.
2. This course introduces the concepts and methods required for the construction of large software intensive systems. It aims to develop a broad understanding of the discipline of software engineering.
3. Capable of team and organizational leadership in computing project settings, and have a broad understanding of ethical application of computing-based solutions to societal and organizational problems.
4. Apply their foundations in software engineering to adapt to readily changing environments using the appropriate theory, principles and processes

UNIT-I:

Software and Software Engineering: The Nature of Software, The Unique Nature of WebApps ,Software Engineering, Software Process, Software Engineering Practice, Software Myths.

Process Models: A Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, The Unified Process, Personal and Team Process Models, Process Terminology, Product and Process.

UNIT-II:

Requirements Analysis And Specification: Requirements Gathering and Analysis, Software Requirement Specification (SRS), Formal System Specification.

Software Design: Overview of the Design Process, How to Characterize of a Design? Cohesion and Coupling, Layered Arrangement of Modules, Approaches to Software Design

UNIT–III:

Function-Oriented Software Design: Overview of SA/SD Methodology, Structured Analysis, Developing the DFD Model of a System, Structured Design, Detailed Design, Design Review, over view of Object Oriented design.

User Interface Design: Characteristics of Good User Interface, Basic Concepts, Types of User Interfaces, Fundamentals of Component-based GUI Development, A User Interface Design

Methodology.

UNIT–IV:

Coding And Testing: Coding, Code Review, Software Documentation, Testing, Unit Testing, Black-Box Testing, White-Box Testing, Debugging, Program Analysis Tool, Integration Testing, Testing Object-Oriented Programs, System Testing, Some General Issues Associated with Testing.

UNIT–V:

Software Reliability And Quality Management: Software Reliability, Statistical Testing, Software Quality, Software Quality Management System, ISO 9000, SEI Capability Maturity Model.

Planning a software project: Effort estimation, project schedule and staffing, quality planning, risk management planning, project monitoring plan, detailed scheduling.

UNIT–VI

Software Maintenance: Software maintenance, Maintenance Process Models, Maintenance Cost, Software Configuration Management.

Software Reuse: what can be Reused? Why almost No Reuse So Far? Basic Issues in Reuse Approach, Reuse at Organization Level.

COURSE OUTCOMES:

1. Define and develop a software project from requirement gathering to implementation.
2. Obtain knowledge about principles and practices of software engineering.
3. Focus on the fundamentals of modeling a software project.
4. Obtain knowledge about estimation and maintenance of software systems.

TEXT BOOKS:

1. Software Engineering A practitioner’s Approach, Roger S. Pressman, Seventh Edition McGraw-Hill International Edition.
2. Fundamentals of Software Engineering, Rajib Mall, Third Edition, PHI.
3. Software Engineering, Ian Sommerville, Ninth edition, Pearson education

REFERENCE BOOKS:

1. Software Engineering : A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
2. Software Engineering, A Precise Approach, PankajJalote, Wiley India,2010.
3. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
4. Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF INFORMATION TECHNOLOGY

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

Year/Semester	**II B. Tech/I Sem**	**L**	**T**	**P**	**C**
Regulation Year	**2020-21**	**3**	**0**	**0**	**3**
Subject	**Discrete Mathematical Structures**				

COURSE OBJECTIVES:

1. To introduce the concepts of mathematical logic.
2. To introduce the concepts of sets, relations, and functions.
3. To perform the operations associated with sets, functions, and relations.
4. To introduce generating functions and recurrence relations.
5. To relate practical examples to the appropriate set, function, or relation model, and interpret the Associated operations and terminology in context. To use Graph Theory for solving problems.

UNIT-I:

Mathematical Logic : Propositional Calculus: Statements and Notations, Connectives, Truth Tables, Tautologies, Equivalence of Formulas, Duality law, Tautological Implications, Normal Forms, Theory of Inference for Statement Calculus, Consistency of Premises, Indirect Method of Proof. Predicate calculus: Predicative Logic, Statement Functions, Variables and Quantifiers, Inference theory for predicate calculus.

UNIT-II:

Number Theory: Number Theory: Properties of Integers, Division Theorem, The Greatest Common Divisor, Euclidean Algorithm, Least Common Multiple, Testing for Prime Numbers, The Fundamental Theorem of Arithmetic, Modular Arithmetic (Fermat’s Theorem and Euler’s Theorem)

UNIT-III:

Set Theory: Introduction, Operations on Binary Sets, Principle of Inclusion and Exclusion. **Relations:** Properties of Binary Relations, Relation Matrix and Digraph, Operations on Relations, Partition and Covering, Transitive Closure, Equivalence, Compatibility and Partial Ordering Relations, Hasse Diagrams.

Functions: Bijective Functions, Composition of Functions, Lattices and its Properties

UNIT-IV:

Combinatorics: Basic of Counting, Permutations, Permutations with Repetitions, Circular

Permutations, Restricted Permutations, Combinations, Restricted Combinations, Generating Functions of Permutations and Combinations, Binomial and Multinomial Coefficients, Binomial and Multinomial Theorems, Pigeonhole Principle and its Application.

UNIT-V:

Graph Theory: Basic Concepts of Graphs, Sub graphs, Matrix Representation of Graphs: Adjacency Matrices, Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian and Hamiltonian Graphs, Multigraphs, Planar Graphs, Euler’s Formula, Graph Colouring and Covering, Chromatic Number, Spanning Trees, Algorithms for Spanning Trees (Problems Only and Theorems without Proofs).

UNIT-VI:

Recurrence Relations: Generating Functions, Calculating Coefficient of Generating Functions, Recurrence Relations, Formulation as Recurrence Relations, Solving Recurrence Relations by Substitution and Generating Functions, Method of Characteristic Roots, Solving Inhomogeneous Recurrence Relations

COURSE OUTCOMES:

1. Ability to apply mathematical logic to solve problems.
2. Understand sets, relations, functions and discrete structures.
3. Able to use logical notations to define and reason about fundamental mathematical concepts such as sets relations and functions.
4. Able to formulate problems and solve recurrence relations.
5. Able to model and solve real world problems using graphs and trees.

TEXT BOOKS :

1. Discrete Mathematical Structures with Applications to Computer Science, J. P. Tremblay and P. Manohar, Tata McGraw Hill.
2. Elements of Discrete Mathematics-A Computer Oriented Approach, C. L. Liu and D. P. Mohapatra, 3rdEdition, Tata McGraw Hill.
3. Discrete Mathematics and its Applicatio-ns with Combinatorics and Graph Theory, K. H. Rosen, 7th Edition, Tata McGraw Hill.

REFERENCE BOOKS:

1. Discrete Mathematics for Computer Scientists and Mathematicians, J. L. Mott, A. Kandel, T.P. Baker, 2nd Edition, Prentice Hall of India.

2. Discrete Mathematical Structures, BernandKolman, Robert C. Busby, Sharon Cutler Ross, PHI.

3. Discrete Mathematics, S. K. Chakraborthy and B.K. Sarkar, Oxford, 2011.

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

Year/Semester	**II B. Tech/I Sem**	**L**	**T**	**P**	**C**
Regulation Year	**2020-21**	**3**	**0**	**0**	**3**
Subject	**Artificial intelligence tools, techniques and applications**				

COURSE OBJECTIVES:

1. Define AI and ML and understand their relationship with data
2. Understand different data wrangling techniques in Python and their significance.
3. Understand different types of supervised learning techniques with their implementation in Python.
4. Intuitively understand basic math fundamental behind each technique
5. Understand performance metrics and tuning performance.
6. Explain the mechanism of unsupervised learning and practice various clustering techniques, Dimensionality reduction and their importance
7. Comprehend text mining and its applications
8. Understand basic working of recommender system
9. Know probabilistic learning models and their applications
10. Understand basics of Artificial and Deep Neural Networks.

UNIT-I:

Intro and basic tools in Python: Introduction to AI and Machine Learning. Emergence of AI. Relationship between AI, ML and Data Science. Types of Machine Learning with definitions and application areas. Types of Data. Data wrangling and manipulation using Numpy and Pandas in Python. Types of data. Data visualization using matplotlib and seaborn.

UNIT-II:

Supervised learning – Regression: Introduction, KNN, Train-Test-Split. k-fold Cross Validation, Linear Regression, Least Squares, Mean Square Error. Plotting regression line and predicting with Scikit Learn. Gradient Descent. Stochastic Gradient Descent. Learning rate. Polynomial Regression in Scikit Learn. Overfitting, Underfitting. Regularization. Measures of accuracy. Bias Vs. Variance trade-off. Hyperparameter tuning.

UNIT-III:

Supervised Learning – Classification: Definition of classification, use cases and algorithms using Scikit Learn, KNN, Logistic Regression, Decision Tree classifier, Support Vector Machines, Naïve

Bayes, Performance measures for classification.

UNIT-IV:

Unsupervised Learning: Introduction, K-Means, Hierarchical clustering techniques. Dimensionality reduction using PCA. Feature Selection and Feature Engineering.

UNIT-V:

Natural Language Processing / Text mining: Introduction. Applications NLP. Components of Natural Language. Steps to get text data into workable format. Term Frequency, Inverse Document Frequency, Bag of Words, ngram, One hot encoding. Notion of corpus. Introduction to NLTK.

UNIT-VI:

Introduction to other common learning methods and applications: Introduction to ANN and deep learning with applications, introduction to ensemble learning with bagging and boosting, random forest and AdaBoost, Recommender Systems-Content and collaborative filtering, Association Rule Mining- Apriori algorithm.

COURSE OUTCOMES:

1. Understand the necessary packages needed for various machine learning algorithms
2. Understand the importance and applications of AI
3. Understand concepts of Machine Learning algorithms and their limitations
4. Understand how to showcase the machine learning outcomes
5. Understand smart solutions for various domains

TEXT BOOKS:

1.Python Machine Learning Cookbook- Practical Solutions from Preprocessing to Deep learning- Chris Albon- O’Reilly publication-First Edition.

2. Introduction to Machine Learning with Python A Guide for Data Scientists, Andreas C. Müller and Sarah Guido, O’Reilly

REFERENCE BOOKS:

1. Python for Data Analysis Data Wrangling with Pandas, NumPy, and IPython, Wes McKinney, O’Reilly

2. Python Data Science Handbook, Essential Tools for Working with Data, Jake VanderPlas, O'Reilly

3. Machine Learning Algorithms, Giuseppe Bonaccorso, Packt Publishing.

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF INFORMATION TECHNOLOGY

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

Year/Semester	**II B. Tech/I Sem**	**L**	**T**	**P**	**C**
Regulation Year	**2020-21**	**3**	**0**	**0**	**3**
Subject	**Data Structures and Algorithms**				

COURSE OBJECTIVES:

1. The fundamental design, analysis, and implementation of basic data structures.
2. Basic concepts in the specification and analysis of programs.
3. Principles for good program design, especially the uses of data abstraction.
4. Significance of algorithms in the computer field.
5. Various aspects of algorithm development.
6. To present different sorting algorithms.

UNIT-I:

Algorithms, Performance analysis- time complexity and space complexity, Asymptotic Notation-Big Oh, Omega and Theta notations, Complexity Analysis Examples. Searching and Sorting: Linear and binary search methods. Bubble sort, Insertion sort, Selection Sort, Quicksort, Merge sort, Heap sort, comparison of sorting methods.

UNIT-II:

Data structures-Linear and non linear data structures, Linear List, Array representation, Linked representation, Vector representation, singly linked lists -insertion, deletion, search operations, doubly linked lists-insertion, deletion operations, circular lists. Representation of single, two-dimensional arrays

UNIT-III:

Stack and Queue, array and linked list representations, infix to postfix conversion using stack, implementation of recursion, Circular queue-insertion, and deletion, Dequeue, array and linked list representations.

UNIT-IV:

Trees- Ordinary and Binary trees terminology, Properties of Binary trees, Binary tree representations, recursive and non-recursive traversals, Inserting a Node into a Threaded Binary Tree, Heaps, Definition of a Max Heap, Insertion into a Max Heap, Deletion from a Max Heap, Priority Queues.

UNIT-V:

Search trees- Binary search tree-Binary search tree, insertion, deletion and searching operations, Balanced search trees, AVL trees-Definition, operations, Red-Black trees –Definition, operations.

UNIT-VI:

Graphs- Introduction, Definition, Graph Representation, Elementary Graph Operations – Vertex Insertion, Vertex Deletion, Edge Insertion, Edge Deletion etc, Depth First Search, Breadth-First Search,

Hashing: Definition, Hash table, Hash function, Collision, Collision Evaluation Techniques-Chaining, Open Addressing.

COURSE OUTCOMES:

1. Provide solutions using data structures
2. Analyze space and time complexity for both iterative and recursive functions.
3. Understand various sorting algorithms and their performance
4. Understand hash functions and collision handling.
5. Identify the right data structure required to solve a problem.

TEXT BOOKS:

1. Data Structures, Using C, Second Edition, Reema Thareja, OXFORD Higher Education.
2. Data structures, Algorithms and Applications in C++, S.Sahni, University Press (India) Pvt.Ltd, 2nd edition, Universities Press Orient Longman Pvt. Ltd.
3. Data structures and Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education. Ltd., Second Edition.
4. Data structures and algorithms in C++, 3rd Edition, Adam Drozdek, Thomson
5. Data structures and Algorithm Analysis in Java, M.A.Weiss, 2nd edition, Addison- Wesley (Pearson Education).

REFERENCES:

1. Introduction to Algorithms, Third Edition, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stien.

2. Data structures and Algorithms in Java, R.Lafore, Pearson education.
3. Data structures and Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education. Ltd., Second Edition.
4. Data structures using C and C++, Langsam, Augenstein and Tanenbaum, PHI

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF INFORMATION TECHNOLOGY

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

Year/Semester	**II B. Tech/I Sem**	**L**	**T**	**P**	**C**
Regulation Year	**2020-21**	**3**	**0**	**0**	**3**
Subject	Digital Logic Design				

COURSE OBJECTIVES:

1. To learn basic tools for the design of digital circuits and fundamental concepts used in the design of digital systems
2. To understand common forms of number representation in digital electronic circuits and to be able to convert between different representations.
3. To implement simple logical operations using combinational logic circuits
4. To design combinational logic circuits, sequential logic circuits
5. To impart to student the concepts of sequential circuits, enabling them to analyze sequential systems in terms of state machines.
6. To implement synchronous state machines using flip flops.

UNIT-I:

Digital Systems and Binary Numbers: Digital Systems, Binary Numbers, Octal and Hexadecimal Numbers, Complements of Numbers, Signed Binary Numbers, Arithmetic addition and subtraction

UNIT-II:

Concept of Boolean algebra: Basic Theorems and Properties of Boolean algebra, Boolean Functions, Canonical and Standard Forms, Minterms and Maxterms,

UNIT-III:

Gate level Minimization: Map Method, Two-Variable K-Map, Three-Variable K-Map, Four Variable K-Maps. Products of Sum Simplification, Sum of Products Simplification, Don't – Care Conditions, NAND and NOR Implementation, Exclusive-OR Function

UNIT-IV:

Combinational Logic: Introduction, Analysis Procedure, Design Procedure, Binary Adder–Subtractor, Decimal Adder, Binary Multiplier, Decoders, Encoders, Multiplexers.

UNIT-V:

Synchronous Sequential Logic: Introduction to Sequential Circuits, Storage Elements: Latches, Storage Elements: Flip-Flops, Analysis of Clocked Sequential Circuits, Mealy and Moore Models of Finite State Machines

UNIT-VI:

Registers and Counters: Registers, Shift Registers, Ripple Counters, Synchronous Counters, Ring Counter, Johnson Counter, Ripple Counter

COURSE OUTCOMES:

1. Able to manipulate numeric information in different forms, e.g. different bases, signed integers, various codes such as ASCII, gray, and BCD.
2. Able to manipulate simple Boolean expressions using the theorems and postulates of Boolean algebra and to minimize combinational functions.
3. Able to design and analyze small combinational circuits and to use standard combinational functions/building blocks to build larger more complex circuits.
4. Able to design and analyze small sequential circuits and devices and to use standard sequential functions/building blocks to build larger more complex circuits.

TEXT BOOKS:

1. Digital Design, 5/e, M.Morris Mano, Michael D Ciletti, PEA.
2. Fundamentals of Logic Design, 5/e, Roth, Cengage.

REFERENCE BOOKS:

1. Digital Logic and Computer Design, M.Morris Mano, PEA.

2. Digital Logic Design, Leach, Malvino, Saha, TMH. 3. Modern Digital Electronics, R.P. Jain, TMH.

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF INFORMATION TECHNOLOGY

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

Year/Semester	II B. Tech/I Sem	L	T	P	C
Regulation Year	2020-21	3	0	0	3
Subject	Object oriented programming through java				

COURSE OBJECTIVES:

1. Gain knowledge about basic Java language syntax and semantics to write Java programs and use concepts such as variables, conditional and iterative execution methods etc.
2. Understand the fundamentals of object-oriented programming in Java, including defining classes, objects, invoking methods etc.
3. Understand the Object-oriented Programming principles like inheritance, polymorphism, and relate these principles to software design.
4. Implementation of Packages, Interfaces and multi-threaded programs in Java.
5. Introduce the concepts of Exception Handling and Files in Java.
6. To Introduce concept of Collections framework.
7. Applying the above concepts for problem solving using Java.

UNIT-I:

Introduction to OOP, procedural programming language and object oriented language, principles of OOP, applications of OOP, history of Java, Java features, JVM, structure of a Java program. Variables, primitive data types, identifiers, literals, operators, expressions, precedence rules and associativity, primitive type conversion and casting, flow of control, arrays, strings, functions, Introduction to lambda expressions.

UNIT-II:

Classes and objects, class declaration, creating objects, methods, method overloading, constructors and constructor overloading, garbage collector, importance of static keyword and examples, this keyword, command line arguments, nested classes.

UNIT-III:

Inheritance, types of inheritance, method overriding, super keyword, final keyword, overriding and abstract class. Interfaces, creating packages, using packages, importance of CLASSPATH and java.lang package, access modifiers. Multithreading: Introduction, Thread life cycle, Creation of threads, Thread priorities, Thread synchronization, Communication between threads

UNIT-IV:

Exception handlings, importance of try, catch, throw, throws, and finally block, user-defined exceptions, assertions. **File I/O:** Reading data from files and writing data to files, random access in a file, accessing data from CSV and Excel files.

UNIT-V:

Arrays, Array vs ArrayList, Strings, StringBuffer, StringBuilder, StringTokenizer. Collections: Introduction to generics, Autoboxing, Overview and hierarchy of Collection framework, List interface, ArrayList, LinkedList, Stack, Queue interface, PriorityQueue, Set interface, HashSet, LinkedHashSet, TreeSet, Collection interface, Iterator interface, Iterable interface, Collections class, Comparable and Comparator interfaces.

UNIT-VI:

Problem Solving: Strings Practice, Arrays Practice, Using Collections, problems using Lists, Priority Queue, Set, Maps, functional filtering and mapping operations on lists with lambdas. Recursion problems.

COURSE OUTCOMES:

Upon successful completion, students will have the knowledge and skills to

1. Write / Read and understand Java-based software code of medium-to-high complexity.
2. Able to understand OOPs concepts, and apply OOPs concepts to solve real world problems.
3. Able to handle Files, and do multi-threaded programs in Java.
4. Able to write robust programs by which handle exceptions.
5. Able to understand and use Collections Framework in Java.
6. Able to apply the above concept to solve real-world problems using appropriate algorithms.

TEXT BOOKS:

1. The complete Reference Java, 11th edition, Herbert Schildt, TMH.

2. Programming in JAVA, Sachin Malhotra, SaurabhChoudary, Oxford.

REFERENCE BOOKS:

1. Introduction to java programming, 7th edition by Y Daniel Liang, Pearson.

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF INFORMATION TECHNOLOGY

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

Year/Semester	II B. Tech/I Sem	L	T	P	C
Regulation Year	2020-21	3	0	0	0
Subject	Quantitative Aptitude-I				

COURSE OBJECTIVES:

1. Understand different number systems, factorization, divisibility and concept of LCM and HCF.
2. Find averages, relation between ratio and proportion, average price of mixture of different quantities and relation between fraction and percentage.
3. Know the concepts of CP,SP, MRP, profit or loss incurred in a transaction.
4. Know the concepts of principal, interest, difference between SI and CI, EMIs.
5. Understand the relation between speed, distance and time for trains and boats in a river.
6. Understand the relation between time and efficiency, combined work and wages paid for the work.

UNIT-I:

Number Systems: Basic number systems –Face and Place Value, Digital sum-Applications, Factors, Multipliers, Prime, & Composite Numbers, Divisibility Rules, LCM and HCF-Remainder Rules.

UNIT-II:

Averages, Ratio& Proportion: Average-Weighted average, Ratio-Concept and properties, Proportions-Mean ,Third and Fourth proportions, Mixtures & Allegations-Definition-Allegation Rule, Percentages-Conversion of Percentages to Fractions and Vice-Versa.

UNIT-III:

Profit& Loss: Cost Price- Selling Price- Marked Price, Discount- Successive Discounts, Profit or Loss Percentage, False Weights- Dishonest Dealer.

UNIT-IV:

Simple & Compound Interest: Principal-Interest Rate-Tenure , Simple Interest-Formula-Sum, Compound Interest-Formula-Relation Between Simple & Compound Interest, loan-EMI, Investments-Shares.

UNIT-V:

Time & Distance: Time-Distance-Speed-Relation, Conversion of Speed , Average Speed, Trains-Relative Speed- Same and Opposite –Platform, Races, Boats-Streams-Upstream and Downstream.

UNIT-VI:

Time & Work: Work-Time-Efficiency, Combined Work-Partnership-Division of Wages, Chain Rule, Pipes and Cisterns-Inlet-Outlet.

COURSE OUTCOMES:

After completing this course, the students will be able to

1. Find number of factors, LCM and HCF of numbers and fractions, least and greatest number divisible by given numbers and leaving some remainder(s).
2. Evaluate average of numbers, Proportions of given ratio, ratio or average price of two quantities of different prices when mixed to get new mix, use relation between fractions and percentages in calculation.
3. Identify the profit or loss incurred in a transaction and how cheating is possible by an unfair trader.
4. Calculate the simple and compound interests ,difference between them and the EMI repayment for a loan.
5. Evaluate the time taken by a train/car for crossing a static object or a moving object and time taken by a person to a row a boat in a river.
6. Calculate the time required for individual or combined work, shares of amount for their work and time taken for a tank/cistern to get filled by inlets and outlet.

TEXT BOOKS:

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

REFERENCES:

1. Arun Sharma, How to Prepare for Quantitative Aptitude for the CAT, Tata McGraw Hill Publishing Company, 2016.
2. Dinesh Khattar, The Pearson Guide to Quantitative Aptitude for Competitive Examinations, Pearson India, 2016

Year/Semester	II B. Tech/I Sem	L	T	P	C
Regulation Year	2020-21	0	0	3	1.5
Subject	Artificial intelligence Tools, Techniques and Applications lab				

COURSE OBJECTIVES:

1) Apply various NumPy and pandas' concepts to preprocess the data.

2) Understand visualization techniques and create reports for machine learning outcomes

3) Understand the importance of statistics in machine learning and apply them on datasets.

4) Understand how to apply machine learning models on data get inference from them.

COURSE OUTCOMES:

1) Preprocess the raw data

2) Apply various statistical methods to understand the data

3) Apply various visualization techniques to understand the data and building reports.

4) Apply Machine learning models on real time data and check the quality of model.

Lab Experiments:

1) **NumPy**

i) Different ways to create NumPy arrays

ii) Add, remove, modify elements in an array.

iii) Arithmetic operations on NumPy array

iv) Slicing and iterating of NumPy arrays

V) Matrix operations onNumPy arrays

2) **Pandas**

i) Create a data Frame manually

ii) Different ways of importing a data frame

iii) Adding, Deleting, Modifying the rows/columns in a dataframe.

iv) Apply functions on dataframe.

V) Iterations on dataframe

vi) Accessing the elements from a dataframe

Vii) Different ways to deal with NA's in dataframe

viii) Groupby operations on dataframe

ix) Merging dataframes

3) Data Visualizations:

i) Line Graphs

ii) Scatter Plots

iii) Histograms

iv) Subplots

v) Joinplots

vi) Heatmaps

4) Basic statistics for machine learning:

Consider a dataset. Apply the following statistical operations on it.

i) Central Tendency- Mean, Median, Mode

ii) Distribution of Data- Range, Interquartile range, Variance, Standard deviation, Correlation.

iii) Draw a box plot to demonstrate Range, Interquartile range.

iv) Show correlation between 2 variables using scatter plot.

V) Draw histogram to show how data is distributed for a given data.

Vi) For the given data, show which attributes as a) continuous b) Ordered c) Binary

5) Prediction

a. Consider a data set and perform univariate linear regression and find the coefficients. Show the relation between independent variable and dependent variable using scatter plot. Show the performance of the model using R-Square error, mean absolute error and Mean Square error

b. Consider a data set and perform multivariate linear regression and find the coefficients. Which attributes are mostly influencing the target variable? Show the performance of the model using R-Square error, mean absolute error and Mean Square error

6) Classification

Consider a datasetand apply following classifiers on it

i) KNN- Classifier

ii) Decision Tree

iii) SVM

iv) Logistic regression

Show the confusion matrix for every model.

Find the accuracy, sensitivity, specificity, F1 score of every model.

Compare the performance of all models

7) Clustering and feature reduction

Consider a dataset and apply the following

i) Apply K-means clustering on the data. Use Elbow method and find the optimal value of K.

ii) Apply Agglomerative clustering on the data. Use dendrograms.

iii) Apply PCA to reduce the number of features in a dataset.

8) Natural Language processing

a. Use NLTK package and perform the following

i) Tokenization

ii) Stemming

iii)Lemmatization

iv) Bag of words

v)TF/IDF

b. Given set of documents, use NLTK to classify them.

9) Ensemble methods.

Consider a dataset

i) Use ensemble method to combine predictive power of decision tree, Logistic regression using bagging technique

ii) Apply ADABOOST on the given data set and draw the confusion matrix for the strong classifier. Apply simple decision tree on the same dataset and compare the performance

iii)) Apply Random Forest on the given data set and draw the confusion matrix for the strong classifier. Apply simple decision tree on the same dataset and compare the performance

10) Association analysis

For a given sales dataset, apply apriori algorithm to generate association rules which is able certain support and confidence.

11) Recommender system.

Use collaborative filtering technique and find similar movies based on the movies watched and rated by a user.

Datasets url's:

1) Kaggle.com

2) https://archive.ics.uci.edu/ml/index.php

3) https://data.gov.in/

4) https://vincentarelbundock.github.io/Rdatasets/datasets.html

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF INFORMATION TECHNOLOGY

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

Year/Semester	**II B. Tech/I Sem**	**L**	**T**	**P**	**C**
Regulation Year	**2020-21**	**0**	**0**	**3**	**1.5**
Subject	**Data Structures and Algorithms lab**				

COURSE OBJECTIVES:

1. To present stacks, queues and their applications
2. To introduce linked lists and its operations.
3. To introduce sorting, searching algorithms.
4. To gain knowledge on trees, graphs and their applications.

COURSE OUTCOMES:

1. Develop ADT necessary for solving applications based on Stacks and Queues.
2. Implement various types of linked lists.
3. Implement various Sorting and searching algorithms.
4. Identify suitable data structures for providing solutions to the real-world problems.

Lab Experiments:

1. Implementation of Searching : (a) Linear Search (b) Binary Search
2. Implementation of Sorting : (a) Bubble Sort (b) Selection Sort
   (c) Insertion Sort(d) Quick Sort (e) Merge Sort
3. Implementation of singly linked list.
4. Implementation of Doubly Linked list.
5. Implementation of Stack using arrays.
6. Implementation of Queue using arrays.
7. Implementation of Converting infix to postfix.
8. Implementation of Stack using linked list.
9. Implementation of Queue using linked list.
10. Implementation of Circular Queue.
11. Implementation of Heap.
12. Implementation of Binary Search Trees.
13. Implementation of Depth First Search technique.
14. Implementation of Breadth First Search technique.
15. Implementation of Dijkstra’s Algorithm.
16. Implementation of Prims Algorithm.
17. Implementation of Kruskal’s Algorithm.

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF INFORMATION TECHNOLOGY

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

Year/Semester	II B. Tech/I Sem	L	T	P	C
Regulation Year	2020-21	0	0	3	1.5
Subject	Oops through Java lab				

COURSE OBJECTIVES:

1. To gain fundamental programming knowledge of OOP
2. To use Exception Handling mechanism in the applications
3. To apply the knowledge of generics and Collections Framework
4. To handle files

COURSE OUTCOMES:

1. Develop Java applications with concepts like Inheritance, Interfaces, packages etc.
2. Implement Exception Handling and Multithreading in Java applications.
3. Develop applications using Collections framework.
4. Read and Write data using different Java I/O streams.

Exercise - 1 (Basics)

a) Write a JAVA program to find the Euclidean distance between two points.

b) Write a java program that displays the roots of a quadratic equation ax^2+bx=0. Calculate the discriminant D and basing on value of D, describe the nature of root.

c) Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers.

Exercise - 2 (Operations, Expressions, Control-flow, Strings)

a) Write a JAVA program to search for an element in a given list of elements using binary search mechanism.

b) Write a JAVA program to sort for an element in a given list of elements using bubble sort.

c) Write a JAVA program to sort for an element in a given list of elements using merge sort.

d) Write a JAVA program using StringBuffer to delete, remove characters.

Exercise - 3 (Arrays)

a) Find smallest number in an array.

b) Find largest number in an array.

c) Count even numbers in an array.

d) Count occurrence of a given number in an array.

e) Check if given number is palindrome or not.

f) Input two arrays and merge them in a new array in ascending order.

g) Find Addition of two 3X3 matrices.

h) Find Multiplication of two 3X3 matrices.

i) Find Transpose of a given matrices.

j) Implement Binary Search.

k) Implement Bubble Sort.

l) Implement Selection Sort.

m) Implement Insertion Sort.

Exercise - 4 (Class, Objects)

a) Write a JAVA program to implement classes – Create a class, methods and invoke them inside the main method.

b) Write a JAVA program to implement a constructor.

Exercise - 5 (Methods)

a) Create a Point class has variables int x and int y. Provide parameterized constructor. Create a class Rectangle. Point p1 is the bottom-left corner and Point p2 is the top-right corner. Write two constructors, one to take Point p1 and Point p2 as arguments and the other to take width and height (in this case (0,0) will be the bottom-left corner). It should have methods to calculate the perimeter() and the area() of the rectangle, move the rectangle by deltax and deltay, find out if a Point p is inside the rectangle or not. Method isInside(Point p). It should also have get methods for both width and height. Write a drive program to test your class.

Exercise - 6 (Inheritance)

a) Create a class Employee and the sub classes Manager and Clerk:

Employee:
Instance Variables: name, empId, salary.
Methods: set and get methods for name, empId, getSalary,setSalary Method

Manager:

Instance Variables: type
Methods: setSalary()

Clerk:
Instance Variables: int speed, int accuacy
Methods: setSalary()

Provide proper constructors for all classes. Create a general class “MyClass”. In this class create objects of Manager, Clerk and Employee class. Set the name, empId and salary attributes for each object, and accordingly display them

Write a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method printArea () that prints the area of the given shape.

b) Definition of a hierarchy of fruits is given below
Fruit contains an abstract method getVitamin() that returns String. Fruit contains a String field color.
Apple’s vitamins are “A B12”.
Banana’s vitamins are “C D”.
Strawberry’s vitamins are “B5 E”.
Blackberry’s vitamins are “C K”.
Apples and bananas grow on trees. All tree fruits provide a void method named peel(). Define class (or interface?) named TreeFruit that has method peel.
Make Apple and Banana extend (or implement?) TreeFruit.
When an Apple is being peeled, it prints out “Peeling an apple.”
GroundFruit.
When a Strawberry is being picked, it prints out “Picking a strawberry.”
i. Implement the classes.
ii. Implement a method named prepareFruits that takes a list of fruits and invokes tree fruits peel method and ground fruits picks methods. i.e. you have to distinguish tree fruits from ground fruits.

```
Public static void prepareFruits (Fruit [] fruits){
 //Implement
}
```

Use the following main method

```
Public static void main(String [] args){
Fruit [] fruits = new Fruit[4];
fruits[0] = new Apple();
fruits [1] = new Banana();
fruits [2] = new Strawberry();
fruits [3] = new Blackberry();
prepareFruits(fruits);
}
```

c) (Shape Hierarchy) Implement the Shape (interface) hierarchy has an abstract function draw() which will be instantiated in concrete classes with what it is drawing. Each abstract class TwoDimensionalShape should contain a method getArea to calculate the area of the two-dimensional shape. Each abstract class ThreeDimensionalShape should have methods getArea and getVolume to calculate the surface area and volume, respectively, of the three dimensional shape. Create a program that uses an array of Shape references to objects of each concrete class in the hierarchy. The program should print a text description of the object to which each array element refers. Also, in the loop that processes all the shapes in the array, determine whether each shape is a Two-DimensionalShape or a ThreeDimensionalShape. If a shape is a TwoDimensionalShape, display its area. If a shape is a ThreeDimensionalShape, display its area and volume.

Shape
2D or 3D
2D – Circle, square
3D – Cube, Sphere

Hint use instanceof to find if it is 2D or 3D.

Your output should appear as follows:

```
Circle: radius: 4
Circle's area is 50
Square: side: 10
```

```
Square's area is 100
Sphere: radius: 2
Sphere's area is 50
Sphere's volume is 33
Cube: side: 8
Cube's area is 384
Cube's volume is 512
```

Exercise - 7 (Runtime Polymorphism)

a) Write a JAVA program that implements runtime polymorphism.

b) Write a JAVA program to create three classes Shape, Circle and Rectangle. Demonstrate runtime polymorphism.

Exercise – 8 (Packages)

a) Write a JAVA program to illustrate CLASSPATH.

b) Write a JAVA program that imports and uses the defined class in your package in the previous problem.

Exercise - 9 (Exception Handling)

a) Design Java Programs that handle the Java built-in Exception to demonstrate exception handling mechanisms.

b) Write a JAVA program to demonstrate multiple catch clauses.

Exercise – 10 (Exception Handling)

a) Write a JAVA program for illustrating throw clause.

b) Write a JAVA program for illustrating finally block.

c) Define an exception called ‘NoMatchException’ that is thrown when a string is not equal to “VITBhimavaram” and Design a Java program that uses this exception.

Exercise – 11 (Strings)

a) Reverse the string

b) Anagram string

c) Count duplicate character

d) Print uppercase & lowercase letters

e) Palindrome String

f) Repeated & non-repeated character

g) Find repeated word in file

h) Reverse words of string object

i) Count the number of vowels

j) Count number of words in string

k) Display vowel, digits & blank spaces

Exercise – 12 (Collections Framework)

a) Write a JAVA program to add, retrieve & remove element from ArrayList

b) Write a JAVA program to Implement LinkedList

c) Write a JAVA program to Sort & reverse the LinkedList elements

d) Write a JAVA program to Implement push() and pop() on Stack

e) Write a JAVA program to display HashTable content

f) Write a JAVA program to search key & value from HashTable

g) Write a JAVA program to remove duplicate key from hashtable

h) Write a JAVA program to copy elements from HashSet to Array

i) Write a JAVA Program to find common elements

j) Write a JAVA Program to insert, retrieve & remove record

k) Write a JAVA Program for binary search

l) Write a JAVA Program to delete duplicate object

m) Write a JAVA Program to implement intersection & union

Exercise – 13 (Collections Framework)

a) Write a JAVA program to implement a stack using LinkedList class.

b) Write a JAVA program to implement a queue using LinkedList class.

c) Write a JAVA program to read a string and print only the unique characters.

d) Write a JAVA program to read a string and print the frequency of each character.

Exercise – 14 (File I/O)

a) Write a JAVA program to read the data from a file and print it on the console.

b) Write a JAVA program to read name, age, and email details and store them in a file.

c) Write a JAVA program to read a CSV file containing marks of students in a class and display the average marks for each subject.

d) Write a JAVA program to read an Excel file containing age of students in a class and display the median age value.

Exercise – 15 (Threads)

a) Write a JAVA program that creates threads by extending Thread class. First thread display “Good Morning“, every 1 sec, the second thread displays “Hello “every 2 seconds and the third display “Welcome” every 3 seconds ,(Repeat the same by implementing Runnable)

b) Write a program illustrating isAlive() and join()

c) Write a Program illustrating Daemon Threads.

Exercise - 16 (Threads)

a) Write a JAVA program for solving Producer-Consumer problem.

b) Write a case study on thread Synchronization after solving the above producer consumer problem.