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 data structures.

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

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

II YEAR I SEMESTER							
S.No	**Subjects**	**L**	**T**	**P**	**C**	**I**	**E**
1	Discrete Mathematical Structures	2	1	-	3	40	60
2	Internet of Things (IoT)	3	-	-	3	40	60
3	Data Structures and Algorithms	3	-	-	3	40	60
4	Computer Organization & Architecture	3	-	-	3	40	60
5	Object Oriented Programming through Java	3	-	-	3	40	60
6	Quantitative Aptitude – I	3	-	-	0	0	0
7	Internet of Things Lab	-	-	3	1.5	40	60
8	Data Structures and Algorithms Lab	-	-	3	1.5	40	60
9	Object Oriented Programming through Java Lab	-	-	3	1.5	40	60
	Total	**17**	**1**	**9**	**19.5**	**320**	**480**
						800	

II YEAR II SEMESTER							
S.No	**Subjects**	**L**	**T**	**P**	**C**	**I**	**E**
1	Software Engineering	3	-	-	3	40	60
2	E-Commerce	3	-	-	3	40	60
3	Database Management Systems	3	-	-	3	40	60
4	Web Technologies	3	-	-	3	40	60
5	Digital Logic Design	3	-	-	3	40	60
6	Logical Reasoning	3	-	-	0	0	0
7	Socially Relevant Project (15 Hrs/Sem)	-	-	1	0.5	20	30
8	Business English Communication Lab	-	-	3	1.5	40	60
9	Design Thinking & Product Innovation Lab	-	-	3	1.5	40	60
10	Database Management Systems Lab	-	-	3	1.5	40	60
11	Web Technologies Lab	-	-	3	1.5	40	60
	Total	**18**	**0**	**13**	**21.5**	**380**	**570**
						950	

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

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

II YEAR I SEMISTER
COMPUTER SCIENCE AND ENGINEERING
R19 SYLLABUS

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

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	2	1	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.

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

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	Internet of Things				

COURSE OBJECTIVES:

1. To understand the fundamentals of the Internet of Things and to know the physical design, logical design and various IoT level models.
2. To teach a student how to design IoT applications and to know the various communication models and protocols.
3. To understand the fundamentals of 8051 Microcontroller and various IoT Platforms.
4. To build a real time IoT application and deploy using Arduino,NodeMCU8266 and Raspberry Pi.
5. To understand various Cloud Computing platforms and Big Data analytics applied in IoT.

UNIT-I :

Introduction to IoT : Introduction to IoT-Characteristics-Physical design - Protocols – Logical design – Enabling technologies – IoT Levels – Domain Specific IoTs.

UNIT-II:

IoT Design and Wireless Communication Protocols : IoT Design Methodology , IoT Components, IoT Design Methodology using home automation and Weather monitoring, Wireless Communication Protocols : 6LoWPAN, Zigbee, WIFI, , Bluetooth and BLE ,LPWANs,Cellular 4G,5G,RFID, Lifi,Widi.

UNIT-III:

8051 Microcontroller and IoT Development Boards : Introduction to Microcontrollers, the 8051 Instruction Set, AT89S8253 Microcontroller, Assembly Language, IoT Development Boards - NodeMCU, ESP8266, Arduino, Intel Galileo and Raspberry Pi.

UNIT-IV:

IoT Protocols : MQTT, UDP, MQTT brokers, publish subscribe modes, HTTP, COAP,XMPP and gateway protocols, IEEE 802.15.4 protocols.

UNIT-V:

Building IoT Applications with Raspberry Pi : Building IoT with RASPBERRY PI- IoT Physical Devices & Endpoints - IoT Device -Building blocks -Raspberry Pi -Board - Linux on Raspberry Pi - Raspberry Pi Interfaces -Programming Raspberry Pi with Python, Introduction to NodeMCU, Arduino and working example.

UNIT-VI:

IoT Platforms, Cloud and Big Data in IoT : Introduction to Cloud computing : Cloud Computing, clouds types and their features, Open Source IoT Platforms, AWS cloud for IoT, ThingSpeak, Python Web Application Framework, Django, AWS web services for IoT.

Introduction to Big Data Analytics : Introduction Big Data, Apache Hadoop framework , Apache Spark and Python Web Application Framework, Django, Data Analytics.

Challenges in IoT and future directions.

COURSE OUTCOMES:

1. Describe IoT basics, different IoT levels and IoT application domains.
2. Describe IoT design methodology and wireless communication protocols.
3. Describe microcontrollers and identify different IoT development boards.
4. Describe application layer and gateway protocols.
5. Describe and use Raspberry Pi to interface with sensors and actuators by writing programs.
6. Gain experience on using IoT with cloud, big data and describe IoT challenges.

TEXT BOOKS:

1. Internet of Things – A hands-on approach, Arshdeep Bahga, Vijay Madisetti, Universities Press, 2015 .

2. IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, Cisco Press, 2017

3. Enabling things to talk – Designing IoT solutions with the IoT Architecture Reference Model, Alessandro Bassi, Martin Bauer, Martin Fiedler, Thorsten Kramp, Rob van Kranenburg, Sebastian Lange, Stefan Meissner, Springer

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

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 COMPUTER SCIENCE & ENGINEERING

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	Computer Organization and Architecture				

COURSE OBJECTIVES:

1. To understand the structure, function, and characteristics of computer systems.
2. To understand the design of the various functional units and components of computers.
3. To identify the elements of modern instruction sets and their impact on processor design.
4. To explain the function of each element of a memory hierarchy.
5. To identify and compare different methods for computer I/O.
6. To Understand computer arithmetic formulate and solve problems, understand the performance requirements of systems
7. To understand the structure of the multiprocessor system

UNIT-I

Basic Structure Of Computers : Computer Types, Functional unit, Basic Operational concepts, Bus structures, Data Representation: complements, Fixed Point Representation. Floating – Point Representation, Error Detection codes.

UNIT-II

Register Transfer Language And Microoperations: Register Transfer language, Bus and memory transfers, Binary adder, Binary adder-subtractor,Arithmetic circuit, logic micro operations, shift micro operations, Arithmetic logic shift unit.

UNI- III

Type of instructions : Data transfer instructions, Arithmetic and Logic instructions,shift and rotate instructions,Branch instructions Stack organization. Instruction formats. Addressing modes.

Computer Arithmetic: Addition,subtraction, Multiplication algorithm, Division algorithm.

UNIT- IV

The Memory Systems : Memory Hierarchy, Main memory, Read Only Memory: ROM,PROM,EPROM,EEPROM,Flash memory, Cache Memory: Mapping techniques, Virtual Memory.

Secondary storage: Auxiliary memory(Magnetic Hard Disks,Optical Disks)

UNIT-V

Input-Output Organization : Peripheral Devices, Input-Output Interface, Asynchronous data transfer:Source initiated data transfer,Destination initiated data transfer,Priority Interrupts : Daisy-chaining interrupt, Parallel priority Interrupt Direct memory Access(DMA).

Design of Control Unit :Hardwired control, Micro programmed control methods.

UNIT-VI

Multi Processors : Introduction, Characteristics of Multiprocessors, Interconnection Structures:(i)Time-shared common bus,(ii)Multi port memory,(iii)8 X 8 Omega switching network ,(iv)Hypercube interconnection, Inter processor Arbitration:Serial and Parallel Arbitration

COURSE OUTCOMES:

1. Students can understand the architecture of modern computer.
2. They can analyze the Performance of a computer using performance equation Understanding of different instruction types.
3. Students can calculate the effective address of an operand by addressing modes.
4. They can understand how computer stores positive and negative numbers.
5. Understanding of how a computer performs arithmetic operation of positive and negative numbers.

TEXT BOOKS

T1. Computer system architecture, M. Morris Mano, 3 rd edition, pearson/phi

T2. Computer organization , carl hamacher, zvonks vranesic, safeazaky, 5 th edition, mcgraw hill.

T3. Computer architecture a quantitative approach, john l. hennessy and david a. patterson, fourth edition elsevier

REFERENCES

1. Computer organization and architecture – william stallings sixth edition, pearson/phi
2. Structured computer organization – andrew s. tanenbaum, 4th edition phi/pearson
3. Fundamentals or computer organization and design, - sivaraama dandamudi springer int. edition

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

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. Understand basic Java language syntax and semantics to write Java programs and use concepts.
2. Apply fundamental object oriented programming concepts to solve problems.
3. Develop Packages, and create Interface and multithreaded applications with synchronization.
4. Apply exception handling and understand file operations.
5. Understand the use of Collection Framework.
6. Apply various concepts of java to solve real world problems.

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 COMPUTER SCIENCE & ENGINEERING

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

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

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	Internet Of Things (IOT) Lab				

Note: Below experiments need to be done with NodeMCU or Arduino or RaspberryPi

List of experiments:

1. Digital Input/output
2. Analog Input/output
3. Using IR Sensor to detect objects.
4. Using LDR
5. PWM application to control LED Brightness
6. Create a localhost server
7. Use NodeMCU to upload free data from Environmental Sensors to Cloud Server
8. Automatically Tweet Sensor Data on Twitter
9. Control Home devices from self-hosted webpage on Amazon AWS
10. Controlling Home Appliance using Google Assistant
11. Calculating Distance using Ultrasonic Sensor
12. Fetching Humidity and Temperature using DHT 11 Sensor
    (Using ThingSpeak, Adafruit and Blynk IoT Platforms)

Project: Home Automation Project

Case Study:

1. Lighting as a service
2. Intelligent Traffic systems
3. Smart Parking
4. Smart water management

VISHNU INSTITUTE OF TECHNOLOGY (AUTONOMOUS) :: BHIMAVARAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

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 COMPUTER SCIENCE & ENGINEERING

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.