Vidya Jyothi Institute of Technology

(An Autonomous Institution)

(Accredited by NAAC & NBA, Approved by AICTE New Delhi & Permanently Affiliated to JNTUH)

Aziznagar Gate, C.B. Post, Hyderabad-500 075

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

B. Tech SYLLABUS

2020

Applicable for students admitted into B.Tech (Regular) from 2020-21

COURSE STRUCTURE & DETAILED SYLLABUS

B.TECH SECOND YEAR COURSE STRUCTURE & SYLLABUS

B.Tech II Year I Semester

S. No.	Category	Course Title	L	T	P	Credits
1	BS	Mathematical and Statistical Foundations	3	0	0	3
2	H&S	Managerial Economics & Financial Analysis	3	0	0	3
3	ES	Computer System Architecture	3	0	0	3
4	PC-1	Data Structures	3	0	0	3
5	PC- 2	Mathematical Foundations of Computer Science	3	0	0	3
6	PC-3	Essentials of Python Programming	3	0	0	3
7	PC Lab - 1	Data Structures lab	0	0	2	1
8	PC Lab - 2	Python Programming lab	0	0	2	1
9	MC-1	Environmental Science/ Gender Sensitization/ Cyber Laws	2	0	0	0
		Total number of Credits				**20**

B.Tech II Year II Semester

S. No.	Category	Course Title	L	T	P	Credits
1	PC-4	Design & Analysis of Algorithms	4	0	0	4
2	PC-5	Operating Systems	3	0	0	3
3	PC-6	Java Programming	3	0	0	3
4.	PC-7	Software Engineering	3	0	0	3
5	PC-8	Database Management Systems	3	0	0	3
6	H&S	Professional Communication	2	0	0	2
7	PC Lab-3	Java Programming Lab	0	0	2	1
8	PC Lab-4	Database Management Systems Lab	0	0	2	1
9	MC-2	Environmental Science/ Gender Sensitization/Cyber Laws	2	0	0	0
		Total number of Credits				**20**

PCSE, JNTUH
(Univ. Nominee)
Pro. B.V. RAMANAIAH

Dr. Jain

V.V.

Computer System Architecture

B.Tech. II Year I Sem.

L	T	P	C
3	0	0	3

Course Outcomes:

- Understand the various number System
- Classify the basics of instruction sets
- Demonstrate the design of functional units of a digital computer system.
- Analyze various computer arithmetic operations.
- Design a pipeline for consistent execution of instruction with minimum hazards

UNIT - I

Number Systems: Binary, Octal, Hex Decimal, and Conversions, range; Binary additions and subtractions (using 1c, and 2c), concept of overflow, representations of negative numbers using 1's and 2's complement and range; BCD numbers: Representation of 8421, 2421, Ex-3, Gray and self-complementary codes; additions and subtractions on 8421 codes.

UNIT - II

Structure Of Computers: Computer types, functional units, basic operational concepts, VonNeumann architecture, bus structures, software, performance, multiprocessors and multicomputer, data representation, fixed and floating point and error detecting codes.
Register Transfer and Micro Operations: Register transfer language, register transfer, bus and memory transfers, arithmetic micro operations, shift microoperations, arithmetic logic shift unit

UNIT III:

Basic Computer Organization and Design: Instruction codes, computer registers, computer instructions, instruction cycle, timing and control, memory reference instructions, input, output and interrupt.
Central Processing Unit: stack organization, instruction formats, addressing modes, data transfer and manipulation, program control, reduced instruction set computer (**RISC**).

UNIT IV:

Computer Arithmetic: Addition and subtraction, multiplication and division algorithms, floating point arithmetic operation, decimal arithmetic unit, and decimal arithmetic operations.

UNIT V:

The Memory System: Basic concepts, semiconductor RAM types of read only memory (ROM), cache memory, performance considerations, virtual memory, secondary storage raid, direct memory access (DMA).
Processor and control unit Basic MIPS implementation Building data path Control Implementation scheme Pipelining Pipelined data path and control Handling Data hazards & Control hazards Exceptions.

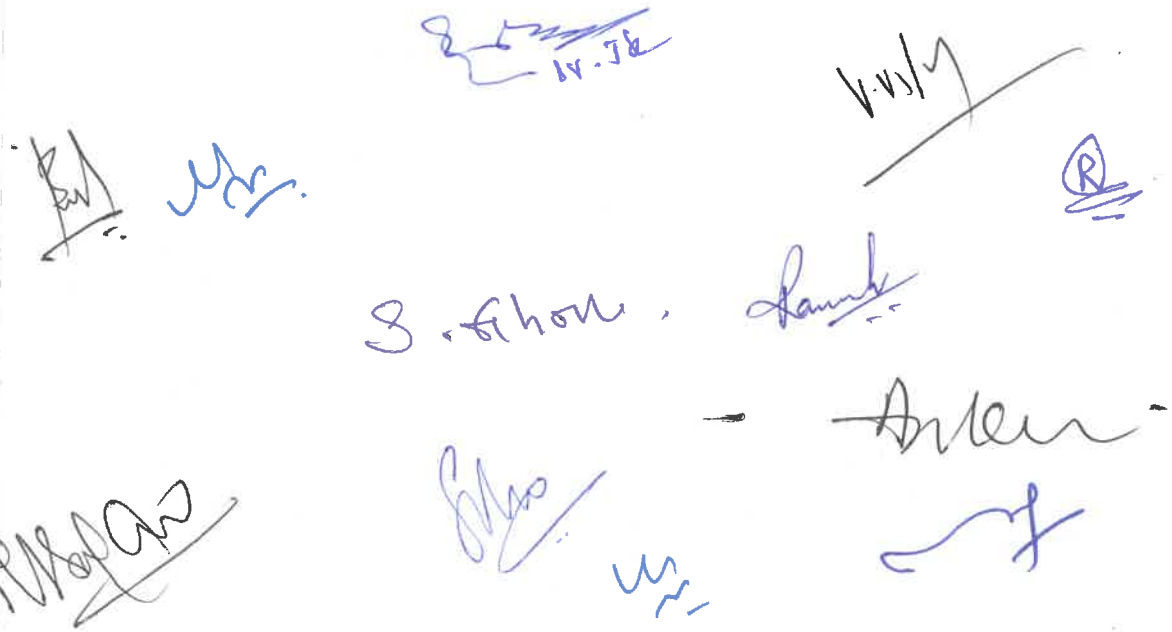
TEXT BOOK:

1. Computer System Architecture – M. Moris Mano, Third Edition, Pearson/PHI.

REFERENCE BOOKS:

1. Computer Organization – Car Hamacher, Zvonks Vranesic, Safea Zaky, Vth Edition, McGraw Hill.

2. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.
Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition, PHI/Pearson.



DATA STRUCTURES

B.Tech II Year I Semester

L	T	P	C
3	0	0	3

Course Outcomes:

1. Understand the concepts of Stacks and Queues with their applications.
2. Analyze various operations on Binary trees.
3. Examine of various concepts of binary trees with real time applications.
4. Analyze the shortest path algorithm on graph data structures.
5. Outline the concepts of hashing, collision and its resolution methods using hash functions.

UNIT -I:

Data Structures: Introduction, Types of data structures, Static and Dynamic representation of data structure and comparison. **Stacks:** Stacks definition, operations on stacks, Representation and evaluation of expressions using Infix, Prefix and Postfix, Algorithms for conversions and evaluations of expressions from infix to prefix and postfix using stack.
Queues: types of Queues- Circular Queue, Deque and operations.

UNIT - II:

Trees: Basic terminologies, Types of Binary Tree: Complete and Full Binary Tree, Extended Binary Trees, Representation of Trees using Arrays and Linked lists (advantages and disadvantages), Tree Traversal, Representation of Algebraic expressions, Threaded Binary Trees.

UNIT - III:

Advanced concepts on trees: Representation and Creation of Binary Search Trees (BST), Operations on BST, Representation and advantages of AVL Trees, algorithms& operations on AVL Trees,Multi-way trees, Definition and advantages of B-trees, B+ Trees, Red-Black Trees.

UNIT - IV:

Graphs-Basic terminology, Representation of graphs: sequential representation, Adjacency, Path Matrix) Linked representation. Graph Traversals-Breadth First Search, Depth First Search algorithms. Spanning Tree, Minimum Spanning Trees- Prim's Algorithm, Kruskals Algorithm, Dijkstra Algorithm.

UNIT -V:

Hashing: General Idea, Hash Functions, collisions, Collision avoidance techniques, Separate Chaining ,Open Addressing-Linear probing, Quadratic Probing, Double Hashing, Rehashing, Extensible Hashing, Implementation of Dictionaries.

Text Books:

1. Data Structures Using C, Second Edition Reema Thereja OXFORD higherEducation
2. Data Structures, A Pseudo code Approach with C, Richard F.Gillberg&Behrouz A. Forouzan, Cengage Learning, India Edition, Second Edition,2005.

Reference Books:

1. Data Structures, Seymour Lipschutz, Schaum's Outlines, Tata McGraw-Hill, Special Second Edition.
2. Data Structures Using C and C++ll, Aaron M. Tenenbaum, YedidyahLangsam and Moshe J. Augenstein PHI Learning Private Limited, Delhilndia.
3. Fundamentals of Data Structuresll, Horowitz and Sahani, *Galgotia Publications* Pvt Ltd Delhi India.
4. Data Structure Using C, A.K. Sharma, Pearson Educationlndia.

MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

B.Tech II Year I Semester

L	T	P	C
3	0	0	3

Course Outcomes:

1. Analyze elementary mathematical arguments.
2. Apply discrete mathematics problems that involve computing permutations and combinations of a set.
3. Analyze problems involving recurrence relations & generating functions.
4. Demonstrate various operations on discrete structures.
5. Apply graph theory models to solve the problems of networks.

UNIT – I:

Foundations: Basics, Sets, Statements, Connectives, Normal Forms, Fundamentals of Logic, Logical Inferences, First order logic and other methods of Proof, Rules of Inference for Quantified Propositions, Automatic Theorem Proving.

UNIT – II:

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumerating Combinations and Permutations with & without repetitions, constrained repetitions, Pigeon hole principle, Inclusion-Exclusion principle.

UNIT – III:

Recurrence Relations: Generating Functions, Calculating coefficient of Generating Function, Solving Recurrence relations by substitution method and Generating Functions, The Method of Characteristic Roots, Solutions to inhomogeneous recurrence relations, Binomial Theorem.

UNIT – IV:

Relations and Digraphs: Relations and Directed Graphs, Special Properties of Binary Relations, Equivalence Relations, Ordering Relations, Lattice, Paths and Closures, Directed Graphs and adjacency matrices.

UNIT – V:

Graphs: Basic Concepts, Isomorphism and Sub-graphs, Trees and Their Properties, Spanning Trees, Binary Trees, Planar Graphs, Euler's Formula, Multi-graphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four-Color Problem.

Text Books:

1. Discrete Mathematics for Computer Scientists and Mathematicians by Joe L. Mott,Abraham Kandel, TheodareP.Baker, Second Edition, PHI, 2009.
2. Discrete Mathematical Structures with Applications to Computer Science, Tremblay J P and Manohar R, Tata McGraw Hill Publishing Company Limited, New Delhi, 2007.

Reference Books:

1. Discrete Mathematics R.K.Bisht, H.S.Dhami, OXFORD Higher Education.
2. Discrete Mathematics and its Applications□, Kenneth H Rosen, Tata McGraw Hill Publishing Company Limited, New Delhi, Sixth Edition, 2007.

ESSENTIALS OF PYTHON PROGRAMMING

B.Tech II Year I Semester

L	T	P	C
3	0	0	3

Course Outcome:

1. Understand the techniques to code in python and write the standard programs using python.
2. Understand to use different IDE's and package of python
3. Understand the python codes for machine learning
4. Having proper idea on different python packages and also the main packages used for machine learning.

Unit I:

Introduction to Python:

Features of Python Language, Data Types, Operators, Expressions, Control Statement, Standard I/O Operations, Functions, OOP using Python, Modules, Packages, Doc Strings, Built-in Functions, Exception, File management.

Unit II:

Strings and Regular Expressions:

String Operations, Built-in String Methods and Functions, Comparing Strings, function in Regular Expression.

Sequence: List, Tuples, Dictionaries.

Why Python for Machine Learning, Other platforms, languages and frameworks for ML

Unit III:

Understanding Python IDEs: Anaconda, Machine learning with scikit-learn, K-means clustering, Data Pre-processing or Data Mugging, Dimensionality Reduction, Entropy, Decision tree as a classifier, Random Forest, Perceptron Learning Algorithm

Unit IV:

Other Python Packages: numpy for matrix computation, iPython for enhanced interactive console, sympy for symbolic calculations, pandas for data structures and analysis, pymc for stochastic calculation, libpgm for Bayesian networks.

Unit V:

Scientific Python using matplotlib, different types of plotting, Graphs, Pie-charts, vector Estimating occupancy using decision tree, Introduction to Theano and Kera.

Text Books

1. "Reema Thareja", Python Programming using Problem Solving Approach, First Edition, Oxford Higher Education,2017.
2. "Sebastian Raschka", Python Machine Learning, PACKT Publishing, Open Source, 2019.

Reference Books:

1. Kenneth A. Lambert, Fundamentals of Python,CENGAGE Learning Custom Publishing,2017.
2. Machine Learning with Python/Scikit-Learn, - Application to the Estimation of Occupancy and Human Activities, GSCOP, Packt Publishing Limited, 2019.

DATA STRUCTURES LAB

B.Tech II Year I Semester

L	T	P	C
0	0	2	1

Course Outcomes:

1. Develop the programs on stacks, trees and its applications.
2. Design and implementation of programs on BST and Graph Traversals.
3. Apply Hashing techniques in real world applications

1. C Programs to illustrate concepts of arrays, structures, unions and enumerated datatypes.
2. Write a Program to implement stack, queue using linked list
3. Program to convert infix to postfix notation
4. Program to evaluate postfix notations
5. Write a program to implement doubly linked list and its operations
6. Program to illustrate tree traversals
    a) In order
    b) Pre order
    c) Post order
7. Program to illustrate insertion, deletion and searching in Binary Search Tree.
8. Program to illustrate Insertion, deletion and Rotation on AVLTrees.
9. Program to illustrate Graph traversals
    i. Breadth First Search
    ii. Depth First Search
10. Program to implement hash table using linear and quadratic probing.
11. Implement Dictionaries using hashing.

PYTHON PROGRAMMING LAB

L	T	P	C
0	0	2	1

B.Tech II Year I Semester

Course Outcomes:

1. Demonstrate the techniques to code in python and write the standard programs using python.
2. Demonstrate to use different IDE's and package of python
3. Design and implement algorithms for Data Science

Exercise 1

a) Installation and Environment setup of python.
b) Write a program to demonstrate the use of basic Data Types
c) Write a program to demonstrate the Operators and Expressions
d) Write a program to demonstrate the Functions and parameter passing Techniques.

Exercise 2

Write a Program to implement

i. Packages ii. Modules iii. Built-in Functions

Exercise 3

Write a Program to implement

a) i. List ii. Tuple iii. Dictionaries
b) Programs on Stings, String Operations and Regular Expressions

Exercise 4

a) Write a Program to implement Class and Object
b) Write a Program to implement Static and Instance methods, Abstract Classes.
c) Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)

Exercise 5

a) Write a program to implement Inheritance and Polymorphism
b) Write a program to implement Files
c) Write a program to illustrate Exception Handling.

Exercise 6

a) Write a program using scikit-learn to implement K-means Clustering
b) Program to calculate the entropy and the information gain
c) Program to implement perceptron learning

Exercise 7

a) Write a program to implement decision tree algorithm and evaluate using accuracy,precison,recall and RMSE.
b) Occupancy estimator using random forest

Exercise 8

a) Calculating with matrices using numpy : inv, pinv, matrix rank, solve, lstsq, svd, transpose, eig, sort, linspace, meshgrid, mgrid, ogrid, concatenate, tile, squeeze, integrate

Exercise 9

a) Program using panda
b) Program using matplotlib use minimum 5 plotting techniques

Exercise 10
a) Graph using matplotlib

Exercise 11
a) Vector using matplotlib

DESIGN AND ANALYSIS OF ALGORITHMS

B.Tech II Year II Semester

L	T	P	C
4	0	0	4

Course Outcomes:

1. Analyze the efficiency of algorithms
2. Develop algorithms divide & conquer, greedy and related problems
3. Examine the performance of Dynamic programming
4. Explain performance of algorithm using Backtracking
5. Analyze NP-Hard and NP-Complete problems

UNIT I:

Introduction: Algorithm, Pseudo code for expressing algorithms, Performance Analysis-Space complexity, Time complexity, Asymptotic Notation- Big oh notation, Omega notation, Theta notation and Little oh notation, Disjoint Sets- disjoint set operations, union and find operations.

Divide and conquer: General method, applications Binary search, Quick sort, Merge sort, Stassen's matrix multiplication.

UNIT II:

Graphs: Breadth First Search, Depth First Search, spanning trees, connected and bi-connected components

Greedy method: General method, Applications- Optimal storage on Tapes, Job sequencing with deadlines, knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

UNIT III:

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

UNIT IV:

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

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

UNIT V:

Lower Bound Theory: Comparison Trees, NP-Hard and NP-Complete problems: Basic concepts, non-deterministic algorithms, NP Hard and NP Complete classes, Satisfiability problem, Clique Decision Problem (CDP), Node cover decision problem.

Text Books:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, SatrajSahni and Rajasekharan, Galgotia publications Pvt.Ltd.

2. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearsoneducation.

References:

1. Introduction to Design and Analysis of Algorithms A strategic approach, R.C.T.Lee, S.S.Tseng, R.C.Chang and T.Tsai, McGrawHill.
2. Data structures and Algorithm Analysis in C++, Allen Weiss, Second edition, Pearsoneducation.

OPERATING SYSTEMS

L	T	P	C
3	0	0	3

B.Tech II Year II Semester

Course Outcomes:

1. Understand the basic functions of Operating systems and system calls.
2. Analyze process scheduling and synchronization.
3. Understand the concepts of memory management.
4. Examine the concepts of MASS storage structure
5. Compare different protection methods of OS and understand the deadlock concepts.

UNIT - I:

Introduction: Operating Systems Objectives and functions, OS Structure, OS Operations, Evolution of Operating Systems - Simple Batch, Multi programmed, time shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, Special - Purpose Systems, Operating System services, user OS Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, Virtual Machines.

UNIT - II:

Process and CPU Scheduling - Process concepts - The Process, Process State, Process Control Block, Threads, Process Scheduling - Scheduling Queues, Schedulers, Context Switching, Preemptive Scheduling, Scheduling Criteria, Scheduling algorithms, Thread scheduling, Case studies: Linux, Windows.

Process Coordination - Process Synchronization, The Critical section Problem, Peterson's solution, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization, Monitors, Case Studies: Linux, Windows.

UNIT - III:

Memory Management and Virtual Memory - Logical & physical Address Space, Swapping, Contiguous Allocation, Paging, Structure of Page Table, Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Performance of Demand Paging, Page Replacement Algorithms, Thrashing.

UNIT - IV:

File System Interface - The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection, File System Implementation - File System Structure, File System Implementation, Allocation methods, Free-space Management, Directory Implementation, Efficiency and Performance.

Mass Storage Structure - Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management.

UNIT - V:

Deadlocks - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery.

Protection - System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection.

TEXT BOOKS:

1. Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne 8th Edition, Wiley Student Edition.

2. Operating systems - Internals and Design Principles, W. Stallings, 6th Edition, Pearson.

REFERENCES BOOKS:

1. Modern Operating Systems, Andrew S Tanenbaum 3rd Edition PHI.
2. Operating Systems A concept - based Approach, 2nd Edition, D. M. Dhamdhere, TMH.
3. Principles of Operating Systems, B. L. Stuart, Cengage learning, India Edition.
4. Operating Systems, A. S. Godbole, 2nd Edition, TMH

JAVA PROGRAMMING

B.Tech II Year II Semester

L	T	P	C
3	0	0	3

Course Outcomes:

1. Understand OOP concepts to apply basic Java constructs.
2. Analyze different forms of inheritance and usage of Exception Handling
3. Understand the different kinds of file I/Multithreading in complex Java programs, and usage of Container classes
4. Contrast different GUI layouts and design GUI applications
5. Construct a full-fledged Java GUI application, and Applet with database connectivity.

UNIT I:

Java Basics History of Java, Java buzzwords, data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program

Fundamentals of Object Oriented Programming: Object-Oriented Paradigm, Basic Concepts of Object Oriented Programming, Applications of OOP. Concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, static keyword, nested and inner classes, Strings, Object class.

UNIT II:

Inheritance & Polymorphism: Introduction, Forms of Inheritance specialization, specification, construction, extension, limitation, combination, Member access rules, super keyword, polymorphism- method overriding, abstract classes, final keyword.

Interfaces and Packages: Introduction to Interfaces, differences between abstract classes and interfaces, multiple inheritance through interfaces, Creating and accessing a package, Understanding CLASSPATH, importing packages.

Exception handling Concepts of exception handling, exception hierarchy, built in exceptions, usage of try, catch, finally, throw, and throws, creating own exception sub classes.

UNIT III:

Files: Introduction to I/O Streams: Byte Streams, Character Streams. File me /O. **Multi-threading:** Differences between multi-threading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication.

Java.util package- Collection Interfaces: List, Map, Set. The Collection classes: Linked List, HashMap, TreeSet, StringTokenizer, Date, Random, Scanner.

UNIT IV:

AWT: Class hierarchy, Component, Container, Panel, Window, Frame, Graphics.

AWT controls: Labels, Button, Scrollbar, Text Components, Checkbox, Checkbox Group, Choice, List, Panes ScrollPane, Dialog and MenuBar.

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapterclasses.

UNIT V:

Layout Manager Border, Grid, Flow, Card and Gridbag.

Applets Concepts of Applets, life cycle of an applet, creating applets, passing parameters toapplets.

JDBC Connectivity: JDBC Type 1 to 4 Drivers, connection establishment, Query Execution.

Text Books:

1. Java- the complete reference, Seventh edition, Herbert Schildt, Tata McGraw Hill.
2. Database Programming with JDBC&JAVA, Second Edition, George Reese, O'Reilly Media.

Reference Books:

1. Thinking in Java Fourth Edition, Bruce Eckel
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
3. Understanding OOP with Java, updated edition, T. Budd, Pearson Education.

SOFTWARE ENGINEERING

L	T	P	C
3	0	0	3

B.Tech II Year II Semester

Course Outcomes:

At the end of the course student would be able to

1. Outline the framework activities for a given project.
2. Examine Right process model for a given project.
3. Analyze various system models for a given Context.
4. Understand various testing techniques for a given project.
5. Identify various risks in project development.

UNIT I:

Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, Software myths.

A Generic view of process: Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), personal and team process models.

UNIT II:

Process Models: The waterfall model, Incremental process models, Evolutionary process model, Unified process model, agile process model.

Software Requirements: Functional and non-functional requirements, the software requirements document.

Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

UNIT III:

System models: Context Models, Behavioral models, Data models, Object models, structured methods.

Design Engineering: Design process and Design quality, Design concepts, the design model, Modeling component level design: design class based components, conducting component level design.

User interface design: Golden rules.

UNIT IV:

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing techniques, Validation testing, System testing.

Product Metrics: Software Quality, Metrics for Analysis Model- function based metrics, Metrics for Design Model-object oriented metrics, class oriented metrics, component design metrics, Metrics for source code, Metrics for Testing, Metrics for maintenance.

UNIT V:

Risk Management: Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

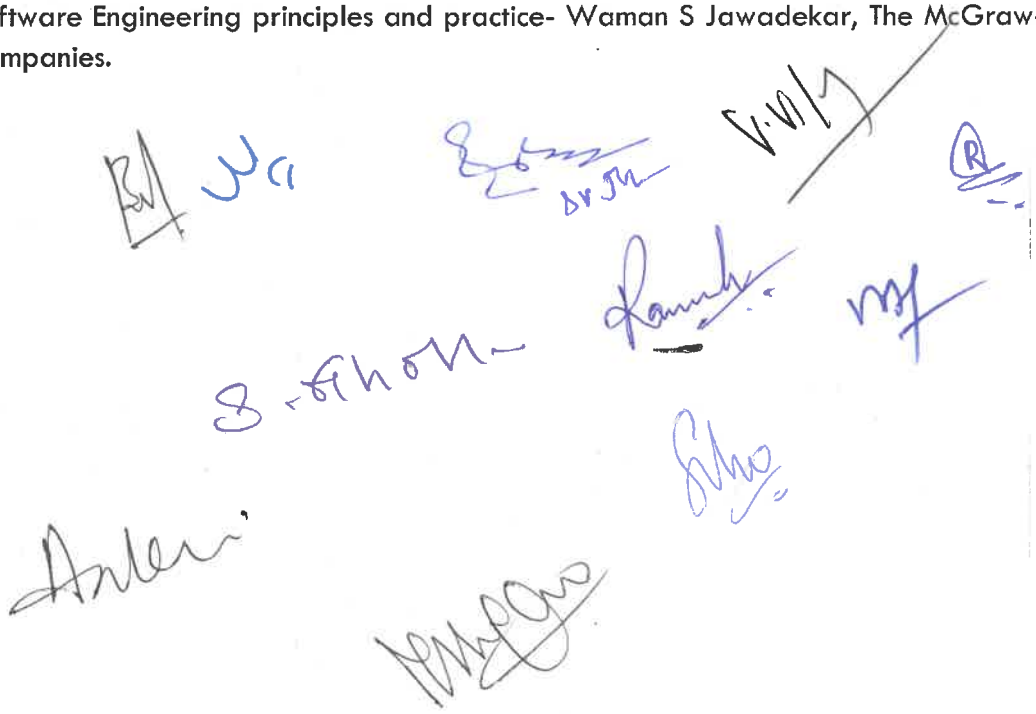
Quality Management: Quality concepts, Software Reviews, Formal technical reviews, Software reliability, The ISO 9000 quality standards.

Text Books:

1. Software Engineering, Apractitioner'sApproach-RogerS.Pressman,6th edition McGraw Hill InternationalEdition.
2. Software Engineering- Sommerville, 7th edition, Pearsoneducation.

References:

1. Software Engineering- K.K. Agarwal&Yogesh Singh, New Age InternationalPublishers
2. Software Engineering, an Engineering approach- James F. Peters, WitoldPedrycz,JohnWiely.
3. Systems Analysis and Design- ShelyCashmanRosenblatt,ThomsonPublications.
4. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.



DATABASE MANGEMENT SYSTEMS

B.Tech II Year II Semester

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the course student would be able to

1. Understand the concepts of Entity-Relationship Model for enterprise level databases.
2. Analyze the database and provide restricted access to different users of database.
3. Understand various Normal forms to carry out schema refinement.
4. Analyze various Concurrency control protocols.
5. Examine working principles of Recovery algorithms

UNIT-I:

Introduction to Database System Concepts: Database-System Applications, Purpose of Database Systems, View, Database Language, Database Architecture, Database Users and Administrators.

Introduction to the Relation Models and Database Design using ER Model: Overview of the Design Process, The Entity-Relationship Model, Constraints, Entity-Relationship Diagrams, Reduction to Relational Schemas, Entity-Relationship Design Issues, Extended E-R Feature, Structure of relational databases , database schema , keys, schema diagrams.

UNIT-II:

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions Nested Sub queries, Modification of theDatabase.

Intermediate and Advanced SQL: Join Expressions, Views, Integrity Constraints, SQL Data Types, Authorization. Functions and Procedures, Triggers.

UNIT-III:

Formal Relational Query Languages: The Relational operations, The Tuple Relational Calculus, The Domain Relational Calculus.

Relational Database Design: Features of Good Relational Designs, Atomic Domains and First Normal Form, Decomposition Using Functional Dependencies, Decomposition Using Multi valued Dependencies, BCNF.

UNIT-IV:

Transactions: Transaction Concept, a Simple Transaction Model, Storage Structure, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Transaction Isolation and Atomicity.

Concurrency Control: Lock-Based Protocols, Deadlock Handling, Timestamp- Based Protocols.

UNIT-V:

Recovery System: Failure Classification, Recovery and Atomicity, Recovery Algorithm, Buffer Management, Failure with Loss of Nonvolatile Storage, ARIES, Remote Backup Systems.

Text Books:

1. Abraham Silberschatz, Henry F. Korth,S.Sudarshan,Database System Conceptsll,6thEdition, Tata McGraw-Hill.
2. RaghuRamaKirshna,JohannesGehrk,DatabaseManagementSystemllTataMcGrawHill 3rd Edition.

Reference Books:

1. PeterRob&CarlosCoronelDatabaseSystemConceptsCengageLearning.
2. RamezElmasri, ShamkanthB. Navrate–FundamentalsofDatabaseSystems7thEdition, Pearson Education.
3. C.J. Date Introduction to Database Systems PearsonEducation

JAVA PROGRAMMING LAB

L	T	P	C
0	0	2	1

B.Tech II Year II Semester

Course Outcomes:

1. Apply basic Java constructs and OOP to solve mathematical problems.
2. Apply Inheritance in Java programs and Analyze Exception Handling code
3. Implement File input/output and multithreading concepts in advanced Java programs.
4. Design different GUI applications using GUI layouts.
5. Apply Applet development and Database connectivity to build GUI applications

Week 1 & 2:

1. Write a program to find total, average of given two numbers by using function with command-line arguments, static datamembers.
2. Write a program to illustrate class and objects.
3. Write a program to illustrate method & constructor overloading.
4. Write a program to illustrate parameter passing using objects.
5. Write a program to illustrate ArrayManipulation.

Week 3:

1. Write a program to illustrate different types of inheritances.
2. Write a java program to illustrate Methodoverriding.
3. Write a java program to demonstrate the concept ofpolymorphism (Dynamic Method Dispatch).
4. Write a program to demonstrate finalkeyword.

Week 4 & 5:

1. Write a program to illustrate the use of creation ofpackages.
2. Write a java program to handle the situation of exceptionhandling using multiple catch blocks.
3. Write a program to implement the concept of User definedExceptions.

Week 6 & 7:

1. Write a program to illustrate Multithreading andMultitasking.
2. Write a program to illustrate threadpriorities.
3. Write a program to illustrate Synchronization

Week 8 & 9:

1. Write a program to implement StringTokenizer.
2. Write a program to read one line at a time, and write it to another file.

Week 10 & 11:

1. Write a program to illustrate Event Handling (keyboard, Mouse events)
2. Write a program to illustrate applet life cycle and parameter passing.

Week 12:

1. Write a program to develop a calculator application using AWT.

Week 13:

1. Write a program to illustrate JDBC.

DATABASE MANAGEMENT SYSTEMS LAB

B.Tech II Year II Semester

L	T	P	C
0	0	2	1

Course Outcomes:

1. Use the SQL commands such as DDL, DML and DCL statements to perform different operations.
2. Apply various Integrity constraints on the database tables.
3. Apply Joins to retrieve the information from multiple tables.
4. Design different Views of tables for different users.
5. Design and implement a PL/SQL program which includes procedures, functions, cursors and triggers.

1. Database Schema for a customer-sale scenario

Customer(**Cust id : integer**, cust_name: string)
Item(**item_id: integer**,item_name: string, price:integer)
Sale(**bill_no: integer**, bill_data: date, **cust_id: integer, item_id: integer**, qty sold: integer)
For the above schema, perform the following—

- **a.** Create the tables with the appropriate integrity constraints
- **b.** Insert around 10 records in each of thetables
- **c.** List all the bills for the current date with the customer names and itemnumbers
- **d.** List the total Bill details with the quantity sold, price of the item and the finalamount
- **e.** List the details of the customer who have bought a product which has aprice>200
- **f.** Give a count of how many products have been bought by eachcustomer
- **g.** Give a list of products bought by a customer having cust_id as5
- **h.** List the item details which are sold as oftoday
- **i.** Create a view which lists out the bill_no, bill_date, cust_id, item_id, price, qty_sold,amount Create a view which lists the daily sales date wise for the last oneweek

2. Database Schema for a Student Libraryscenario

Student(**Stud_no : integer**,Stud_name: string)
Membership(**Mem_no: integer**,**Stud_no: integer**)
Book(**book_no: integer**, book_name:string, author: string)
Iss_rec(**iss_no:integer**, iss_date: date, **Mem_no: integer, book_no: integer**)
For the above schema, perform the following

- **a.** Create the tables with the appropriate integrityconstraints
- **b.** Insert around 10 records in each of thetables
- **c.** List all the student names with their membershipnumbers
- **d.** List all the issues for the current date with student and Booknames
- **e.** List the details of students who borrowed book whose author isCJDATE
- **f.** Give a count of how many books have been bought by eachstudent
- **g.** Give a list of books taken by student with stud_no as5

h. List the book details which are issued as oftoday
i. Create a view which lists out the iss_no, iss _date, stud_name, bookname
j. Create a view which lists the daily issues-date wise for the last oneweek

3. Database Schema for a Employee-payscenario

employee(**emp_id:**integer,emp_name:string)
department(**dept_id:integer**,dept_name:string)
paydetails(**emp_id : integer,dept_id: integer,** basic: integer, deductions: integer, additions: integer, DOJ: date)
payroll(**emp_id : integer,** pay_date: date)
For the above schema, perform the following

a. Create the tables with the appropriate integrityconstraints
b. Insert around 10 records in each of thetables
c. List the employee details departmentwise
d. List all the employee names who joined after particulardate
e. List the details of employees whose basic salary is between 10,000 and20,000
f. Give a count of how many employees are working in eachdepartment
g. Give a names of the employees whosenetsalary>10,000
h. List the details for anemployee_id=5
i. Create a view which lists out the emp_name, department, basic, dedeuctions,netsalary
j. Create a view which lists the emp_name and hisnetsalary

4. Database Schema for a Video Library scenario

Customer(**cust_no: integer**,cust_name: string)
Membership(**Mem_no: integer**, **cust_no: integer**)
Cassette(**cass_no:integer**, cass_name:string, Language:String)
Iss_rec(**iss_no: integer**, iss_date: date, **mem_no; integer, cass_no: integer**)
For the above schema, perform the following

a. Create the tables with the appropriate integrityconstraints
b. Insert around 10 records in each of thetables
c. List all the customer names with their membershipnumbers
d. List all the issues for the current date with the customer names and cassettenames
e. List the details of the customer who has borrowed the cassette whose title is —The Legendll
f. Give a count of how many cassettes have been borrowed by eachcustomer
g. Give a list of book which has been taken by the student with mem_no as5
h. List the cassettes issues fortoday
i. Create a view which lists outs the iss_no, iss_date, cust_name,cass_name
j. Create a view which lists issues-date wise for the last oneweek

5. Database Schema for a student-Lab scenario

Student(**stud_no: integer**, stud_name: string, **class: string**)
Class(**class: string,descrip:string**)

Lab(**mach_no: integer**, Lab no: integer, description: String)
Allotment(**Stud_no: Integer, mach_no: integer, day of week: string**)
For the above schema, perform the following

a. Create the tables with the appropriate integrityconstraints
b. Insert around 10 records in each of thetables
c. List all the machine allotments with the student names, lab and machinenumbers
d. List the total number of lab allotments daywise
e. Give a count of how many machines have been allocated to the 'IT'class
f. Give a machine allotment details of the stud_no 5 with his personal and classdetails
g. Count for how many machines have been allocated in **Lab_no 1** for the day of the week as –Monday‖
h. How many students class wise have allocated machines in thelabs
i. Create a view which lists out the stud_no, stud_name, mach_no, lab_no,dayofweek
j. Create a view whichliststhe machine allotmentdetailsfor–Thursday.

6. Create a cursor, which displays all employee numbers and names from the EMP table.
7. Create a cursor, which update the salaries of all employees as per the given data.
8. Create a cursor, which displays names of employees having salary > 50000.
9. Create a procedure to find reverse of a given number
10. Create a procedure to update the salaries of all employees as per the given data
11. Create a procedure to demonstrate IN, OUT and INOUT parameters
12. Create a function to check whether given string is palindrome or not.
13. Create a function to find sum of salaries of all employees working in depart number 10.
14. Create a trigger before/after update on employee table for each row/statement.
15. Create a trigger before/after delete on employee table for each row/statement.
16. Create a trigger before/after insert on employee table for each row/statement.