VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY HYDERABAD
B.TECH. II YEAR
COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

III SEMESTER **A19**

Course Code	Title of the Course	L	T	P/D	Contact Hours/ Week	Credits
A19PC1CS01	Digital Logic Design	2	1	0	3	3
A19PC1CS02	Mathematical Foundations for Computer Science	2	1	0	3	3
A19PC1IT02	Java Programming	2	1	0	3	3
A19PC1IT03	Computer Organization	3	0	0	3	3
A19PC1CS06	Software Engineering	3	0	0	3	3
A19PC2IT03	Java Programming Laboratory	0	0	3	3	1.5
A19PC2CS02	Software Engineering Laboratory	0	0	3	3	1.5
A19PC2IT02	Python Programming Laboratory	0	0	2	2	1
Total		**12**	**3**	**8**	**23**	**19**
A19MN6HS02	Environmental Science	2	0	0	2	0

IV SEMESTER **A19**

Course Code	Title of the Course	L	T	P/D	Contact Hours/ Week	Credits
A19BS1MT11	Probability, Statistics and Queuing Theory	3	0	0	3	3
A19HS1MG02	Engineering Economics and Accountancy	3	0	0	3	3
A19PC1CS05	Formal Language and Automata Theory	2	1	0	3	3
A19PC1CS31	Operating Systems using Linux	3	0	0	3	3
A19PC1CS03	Design and Analysis of Algorithms	3	0	0	3	3
A19PC1CS04	Database Management Systems	3	0	0	3	3
A19PC2CS31	Operating Systems Using Linux Laboratory	0	0	3	3	1.5
A19PC2CS01	Database Management Systems Laboratory	0	0	3	3	1.5
Total		**17**	**1**	**6**	**24**	**21**

L – Lecture T – Tutorial P – Practical D – Drawing

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. III Semester – CSE & IT	**L**	**T/P/D**	**C**
	2	**1**	**3**

(A19PC1CS01) DIGITAL LOGIC DESIGN

COURSE OBJECTIVES:

- To analyze and explore uses of logic functions for building digital logic circuits
- To explore the combinational logic circuits
- To examine the operation of sequential (synchronous and asynchronous) circuits
- To understand the programming concepts of HDL for simulating any type of logic circuits

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Simplify the complex logic functions using k-maps and tabulation methods
CO-2: Build any type of combinational circuits that help in further designing memory elements
CO-3: Design Synchronous and Asynchronous sequential circuits using memory elements
CO-4: Apply the concepts of HDL for simulating the logic functions, combinational and sequential circuits

UNIT – I:
Numbers Systems and Codes: Review of number systems, number base conversion, binary arithmetic, binary weighted and non-weighted codes, Complements, Signed binary numbers, Error Detection and Correcting Codes, Binary Logic.

UNIT – II:
Boolean Algebra and Gate Level Minimization: Postulates and theorems, representation of switching functions, SOP and POS forms –Canonical forms, digital logic gates, Karnaugh Maps –minimization using three variable, four variable and five variable K-Maps, Don't Care Conditions, NAND and NOR implementation , Other Two-Level Implementation, Exclusive-OR function, Integrated Circuits, Hardware Description Language(HDL).

UNIT – III:
Design of Combinational Circuits: Combinational Circuits- Analysis and Design Procedure, Binary adder and subtractors, Binary multiplier, magnitude comparator, BCD adder, Decoders, Encoders, Multiplexers, Demultiplexers, HDL for Combinational Circuits.

UNIT – IV:
Design of Sequential Circuits: Combinational Vs Sequential Circuits, Latches, Flip Flops-RS flip flop, JK flip flop, T flip flop, D flip flop, Master-Slave Flip flop- Flip Flops excitation functions, Conversion of one flip flop to another flip flop, Asynchronous Vs Synchronous circuits, Analysis of clocked sequential circuits, State Table, State Diagram, State Reduction and State Assignment, Mealy and Moore Machines, HDL for Sequential circuits.

UNIT – V:

Counters and Registers: Design of synchronous counters, Ripple Counters, Asynchronous counters, Registers, Shift Registers, HDL for counters and registers.

Memory: Random Access Memory, Read Only Memory, Programmable Logic Array, Programmable Array Logic.

UNIT – VI:

Asynchronous Sequential Logic: Introduction, Analysis Procedure, Circuits with Latches, Design Procedure, Reduction of state and flow Tables, Race Free State Assignment, Hazards, Design examples.

TEXT BOOKS:

1. Digital Design, M. Morris Mano, 3rd Edition, Pearson Education/PHI
2. Switching and Finite Automata Theory, ZviKohavi, Tata McGraw-Hill

REFERENCES:

1. Fundamentals of Logic Design, Roth, 5th Edition, Thomson
2. Switching and Logic Design, C. V. S. Rao, Pearson Education
3. Digital Principles and Design, Donald D. Givone, Tata McGraw-Hill
4. Fundamentals of Digital Logic & Micro Computer Design, M. Rafiquzzaman, 5th Edition, John Wiley

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. III Semester – CSE & IT	**L**	**T/P/D**	**C**
	2	**1**	**3**

(A19PC1CS02) MATHEMATICAL FOUNDATIONS FOR COMPUTER SCIENCE

COURSE OBJECTIVES:
- To apply logical reasoning to a variety of problems
- To understand the concepts on elementary combinatorics and permutations
- To analyze the properties of graphs and trees
- To evaluate various methods for solving the recurrence relations

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Demonstrate problems using statement calculus, predicate logic and set theory
CO-2: Apply and calculate permutations and combinations
CO-3: Understand the use of graphs and trees as models
CO-4: Solve various problems using recurrence relation techniques

UNIT – I:
Mathematical Logic: Statements and notations, Connectives, Well-formed formulas, Truth Tables, tautology, equivalence implication, Normal forms, Rules of inference, Consistency, proof of contradiction, Automatic Theorem Proving.
Predicates: Quantifiers, Predicative logic, Free & Bound variables.

UNIT – II:
Set Theory: notations, inclusion and equality sets, operations on sets, Venn diagrams.
Relations: Properties of binary Relations, equivalence, transitive closure, compatibility and partial ordering relations, Hasse diagram.
Functions: Types of Functions, Inverse Function, Composition of functions, recursive Functions.

UNIT – III:
Elementary Combinatorics: Basics of counting, Combinations & Permutations, with repetitions, Constrained repetitions, the principles of Inclusion – Exclusion, Pigeon hole principle.

UNIT – IV:
Graphs: Graphs and their Properties, Degree, Connectivity, Path, Cycle, Sub graph, Isomorphism, Eulerian and Hamiltonian Walks, Planar Graphs, Graph coloring, Chromatic Numbers.

UNIT – V:
Trees: Properties of trees – Distance and centers in tree – Rooted and binary trees. Spanning trees, BFS, DFS, Spanning trees in a weighted graph.

UNIT – VI:
Recurrence Relations: Generating Functions, Function of Sequences, Calculating Coefficients of generating functions, Recurrence relations, Solving recurrence relation by substitution and Generating functions, the method of Characteristic roots, solution of Inhomogeneous Recurrence Relations.

TEXT BOOKS:

1. Discrete Mathematical Structures with Applications to Computer Science, J. P. Trembly and R. Manohar, Tata McGraw-Hill
2. Discrete Mathematics for Computer Scientists & Mathematicians, J. L. Mott, A. Kandel, T. P. Baker, 2nd Edition, PHI
3. Graph Theory: With Application to Engineering and Computer Science, Narsingh Deo, Prentice Hall of India, 2003

REFERENCES:

1. Elements of Discrete Mathematics, A Computer Oriented Approach, C. L. Liu and D. P. Mohapatra, 3rd Edition, Tata McGraw-Hill
2. Discrete Mathematics and its Applications, Kenneth H. Rosen, Tata McGraw-Hill
3. Discrete Mathematics, Norman L. Biggs, 2nd Edition, Oxford University Press

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. IV Semester – CSE & IT	**L**	**T/P/D**	**C**
	2	**1**	**3**

(A19PC1IT02) JAVA PROGRAMMING

COURSE OBJECTIVES:
- To understand fundamental concepts and constructs of Java
- To implement Different object-oriented Concepts in Java
- To develop the concepts of Multi-Threading and IO-Streams
- To establish connection to the databases

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Write Java programs using various programming constructs using Java
CO-2: Solve different mathematical problems using OOP Paradigm
CO-3: Understand and use Java Collection Framework
CO-4: Design and analyze the solutions for Thread and database connectivity concepts

UNIT – I:
Java Evolution: Java Features - How Java differs from C and C++ - Java and Internet - Java and World Wide Web - Web Browsers - Hardware and Software Requirements - Java Environment. Overview of Java Language: Simple Java Program - Java Program Structure - Java Tokens- Java Statements - Implementing a Java Program - Java Virtual Machine - Constants - Variables - Data types - Scope of Variables-Symbolic Constants-Type Casting and type promotions – Operators, Operator Precedence and Associativity - Control Statements – break - continue- Arrays-Multi dimensional arrays, Wrapper Classes - Simple examples.

UNIT – II:
Classes and Objects - Constructors – methods - this keyword – garbage collection-finalize - Overloading methods and constructors - Access Control- Static members – nested and inner classes – command line arguments - variable length arguments.
Inheritance: types of inheritance, benefits of inheritance. super keyword, Polymorphism, dynamic method dispatch –abstract classes – exploring String class.

UNIT – III:
Packages and Interfaces: Defining and accessing a package – understanding CLASSPATH – access protection importing packages – Interfaces - Defining and implementing an interface, Applying interfaces
Exception Handling-Fundamentals, usage of try, catch, multiple catch clauses, throw, throws and finally. Java built in Exceptions and creating user defined exceptions.

UNIT – IV:
The Collection Framework: Collection Objects – Sets, Lists, Queues, Maps – ArrayList-LinkedList - Vector– HashSet– LinkedHashSet– TreeSet– HashMap– Hashtable. Retrieving Elements from Collections – Enumeration, Iterator, List Iterator – String Tokenizer and Arrays Class – Sorting using Comparable and Comparator.

UNIT – V:

Multithreaded Programming: Java Thread life cycle model – Thread creation - Thread Exceptions - Thread Priority – Synchronization - Runnable Interface - Interthread Communication - Deadlock - Suspending, Resuming and stopping threads.

I/O Streams: File – Streams – Advantages - The stream classes – Byte streams – Character streams, Serialization, File Class and Methods.

UNIT – VI:

JDBC: JDBC Architecture, JDBC – ODBC Connectivity Steps, Connectivity steps with mysql database, Statement, PreparedStatement, CallableStatement, ResultSet, ResultSetMetaData, DatabaseMetaData, Transaction Management, Batch Processing, RowSet Interface. REST API's.

TEXT BOOKS:

1. The Complete Reference Java J2SE, Herbert Schildt, 5th Edition, TMH Publishing Company Ltd., New Delhi
2. Big Java, Cay Horstmann, 2nd Edition, John Wiley and Sons

REFERENCES:

1. Java How to Program, H. M. Dietel and P. J. Dietel, 6th Edition, Pearson Education/PHI
2. Core Java 2, Vol. 1, Fundamentals, Cay S. Horstmann and Gary Cornell, 7th Edition, Pearson Education
3. Core Java 2, Vol. 2, Advanced Features, Cay S. Horstmann and Gary Cornell, 7th Edition, Pearson Education

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. IV Semester	**L**	**T/P/D**	**C**
	3	**0**	**3**

(A19PC1IT03) COMPUTER ORGANIZATION

COURSE OBJECTIVES:

- To describe the functional blocks of a computer to interpret the instructions and various addressing modes for the execution of instruction cycle
- To perform Arithmetic micro operations on integers and floating-point numbers
- To analyze the cost performance and design trade-offs in designing and constructing a computer processor including memory
- To discuss the different ways of communicating with I/O devices & interfaces and the design techniques to enhance the performance using pipelining, parallelism

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Interpret the functional architecture of computing systems
CO-2: Explore memory, control and I/O functions
CO-3: Impart the knowledge on micro programming
CO-4: Analyze instruction level parallelism, Concepts of advanced pipeline techniques

UNIT – I:
Functional Blocks of a Computer: CPU, memory, input-output subsystem, control unit. Instruction set architecture of a CPU – registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set.
Case study – Instruction set of some common CPUs

UNIT – II:
Data Representation: Signed number representation, fixed and floating point representations, character representation.
Computer Arithmetic: Integer Addition and Subtraction - Ripple carry adder, carry look- ahead adder. Multiplication – Shift-and add, Booth multiplier, carry save multiplier. Division – Restoring and non-restoring techniques, floating point arithmetic.

UNIT – III:
Microprogrammed Control: Control memory, address sequencing, micro program example, and design of control unit, hardwired control, and micro programmed control.

UNIT – IV:
Memory System Design: Semiconductor memory technologies. SRAM vs DRAM.
Memory Organization: Memory interleaving, concepts of hierarchical memory organization, cache memory, cache size vs block size, mapping functions, replacement algorithms, write policies, virtual memory, secondary storage.

UNIT – V:
Peripheral Devices and their Characteristics: Input-output subsystems, I/O device interface, I/O transfers, - program controlled, Interrupt driven and DMA, privileged and non –privileged instructions, software interrupts and exceptions. Programs and

processes – role of interrupts in process state transitions, I/O device interfaces – SCSI, USB.

UNIT – VI:

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction pipe line, RISC pipeline Vector Processing, Array Processors

TEXT BOOKS:

1. Computer Organization and Design: The Hardware/Software Interfaces, David A. Patterson and John L. Hennessy, 5th Edition, Elsevier
2. Computer Organization and Embedded Systems, Carl Hamacher, 6th Edition, McGraw Hill Higher Education

REFERENCES:

1. Computer System Architecture, M. Morris Mano, 3rd Edition
2. Computer Architecture and Organization, John P. Hayes, 3rd Edition, WCB/McGraw-Hill
3. Computer Organization and Architecture: Designing for Performance, William Stallings, 10th Edition, Pearson Education
4. Computer System Design and Architecture, Vincent P. Heuring and Harry F. Jordan, 2nd Edition, Pearson Education

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. IV Semester – CSE & IT	**L**	**T/P/D**	**C**
	3	**0**	**3**

(A19PC1CS06) SOFTWARE ENGINEERING

COURSE OBJECTIVES:

- To identify the importance of software engineering principles and software process framework
- To understand contemporary approaches for design model and requirements validation
- To explore various metrics and quality assurance strategies
- To analyse different strategies for testing and risk management

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Analyse software engineering framework activities and process models that can be tailored with appropriate methods for developing the projects

CO-2: Design relevant software system models from the available software requirements and validate desired user model with realistic constraints

CO-3: Deliver quality software products by applying software testing strategies and product metrics over the entire system life cycle

CO-4: Specify contemporary issues of handling risk management in Software development

UNIT – I:

Introduction to Software Engineering: Software characteristics, changing nature of software, software myths.

A Generic View of Process: Software engineering-A layered technology, process frame work, The Capability Maturity Model Integration (CMMI)

UNIT – II:

Process Models: The water fall model, Incremental process models, evolutionary process models, agile process

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:

Modeling with UML: Modeling Concepts and Diagrams - Use Case Diagrams - Class Diagrams - Interaction Diagrams - State chart Diagrams – Activity Diagrams - Package Diagrams - Component Diagrams – Deployment Diagrams -Diagram Organization-Diagram Extensions.

UNIT – IV:

Design Engineering: Design process and design quality, design concepts, design model.

Testing Strategies: A strategic approach to software testing, Testing Strategies, Black box and White box testing.

UNIT – V:

Product Metrics: Metrics for analysis model, Metrics for design model, Metrics for source code, Metrics for testing, Metrics for maintenance

Metrics for Process and Projects: Software measurement, Metrics for software quality

UNIT – VI:

Risk Management: Reactive vs. Proactive risk strategies, Software risks, Risk identification, Risk projection, RMMM plan

Quality Management: Quality concepts, Software quality assurance, Formal technical reviews, ISO 9000 Quality standards.

TEXT BOOKS:

1. Software Engineering - A Practitioner's Approach, Roger S. Pressman, McGraw-Hill International Edition, 6th Edition, 2001
2. Software Engineering, Ian Sommerville, 7th Edition, Pearson Education Asia, 2000
3. The Unified Modeling Language User Guide, Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education

REFERENCES:

1. An Integrated Approach to Software Engineering, Pankaj Jalote, Springer Verlag, 1997
2. Software Engineering – An Engineering Approach, James F. Peters and Witold Pedryez, John Wiley and Sons, New Delhi, 2000
3. Software Engineering Fundamentals, Ali Behforooz and Frederick J. Hudson, Oxford University Press, New Delhi, 1996

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. IV Semester – CSE & IT	**L**	**T/P/D**	**C**
	0	**3**	**1.5**

(A19PC2IT03) JAVA PROGRAMMING LABORATORY

COURSE OBJECTIVES:

- To write the Java Programs related to classes and methods
- To build Solutions for exceptions and basic I/O streams
- To develop solid Java programming skills and the ability to design simple case studies
- To implement the concepts of object oriented to develop a real world application

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Analyze and design a computer program to solve real world problems based on object- oriented principles

CO-2: Implement concurrent programming using Multithreading concepts

CO-3: Identify appropriate Collection classes in problem solving

CO-4: Establish connection to the database using Java

WEEK 1:

1. Write a Java program to print all the twin primes below 1000. (A twin prime is a prime number that differs from another prime number by two. (3, 5), (5, 7), (11, 13), (17, 19), (29, 31), (41, 43), .821, 823), etc. .
2. Write a Java program to implement matrix multiplication. (Take the input from keyboard).
3. Write a Java program for sorting a given list of names in ascending order.

WEEK 2:

4. The Fibonacci sequence is defined by the following rule. The first two values in the sequence are 1 and 1. Every subsequent value is the run of the two values preceding it. Write a Java program that uses both recursive and non-recursive functions to print the nth value in the Fibonacci sequence.
5. Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.

WEEK 3:

6. Write a Java program that checks whether a given string is a palindrome or not from command line. Ex: MALAYALAM is a palindrome.
7. Write a Java program that prints all real solutions to the quadratic equation ax2 + bx + c = 0. Read in a, b, c and use the quadratic formula. If the discriminant b2 - 4ac is negative, display a message stating that there are no real solutions.
8. Write a Java program to implement constructor overloading.

WEEK 4:

9. Write a Java program to implement variable length arguments
10. Write a Java program to implement the use of inner classes.

WEEK 5:

11. Write a Java program to implement dynamic method dispatch.

12. Write a Java program that illustrates how run time polymorphism is achieved.

WEEK 6:

13. Write a Java program that illustrates the following
    a) Creation of simple package.
    b) Accessing a package.
    c) Implementing interfaces.
14. Write a Java program that illustrates built in exceptions.
15. Write a Java program to throw an exception "Insufficient Funds" while withdrawing the amount in the user account.

WEEK 7:

16. Write a Java program for creating multiple threads
    a. Using Thread class b. Using Runnable interface
17. Write a Java program for creating multiple threads. The main method sleeps for 10 seconds at the end of which all the threads should be terminated.

WEEK 8:

18. Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.

WEEK 9:

19. Write a Java program to create a file and write data into the file using Character Stream.
20. Write a Java program that reads on file name from the user then displays information about whether the file exists, whether the file is readable, whether the file is writable, the contents of file and the length of the file in bytes.

WEEK 10:

21. Write a Java program to perform the following operations on ArrayList, LinkedList, HashSet.
    a) Insertion
    b) Deletion
    c) Retrieval

WEEK 11:

22. Write a program to store Employee objects in a TreeSet and sort the objects based on employee salary using Comparator/Comparable.

WEEK 12:

23. Write a Java program to establish the connection to the database and perform the following operations.
    a) Retrieval
    b) Insertion
    c) Deletion
24. Write a Java program to call the stored procedure from a database.

WEEK 13:

25. Explore REST APIs.

TEXT BOOKS:

1. The Complete Reference Java J2SE, Herbert Schildt, 5th Edition, TMH Publishing Company Ltd., New Delhi
2. Big Java, Cay Horstmann, 2nd Edition, John Wiley and Sons

REFERENCES:

1. Java How to Program, H. M. Dietel and P. J. Dietel, 6th Edition, Pearson Education/PHI
2. Core Java 2, Vol. 1, Fundamentals, Cay S. Horstmann and Gary Cornell, 7th Edition, Pearson Education
3. Core Java 2, Vol. 2, Advanced Features, Cay S. Horstmann and Gary Cornell, 7th Edition, Pearson Education

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. III Semester – CSE & IT	**L**	**T/P/D**	**C**
	0	**3**	**1.5**

(A19PC2CS02) SOFTWARE ENGINEERING LABORATORY

COURSE OBJECTIVES:

- To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: To translate end-user requirements into system and software requirements
CO-2: To generate a high-level design of the system from the software requirements
CO-3: To experience and/or to have awareness of testing problems and will be able to develop a simple testing report

LIST OF EXPERIMENTS:

Do the following 7 exercises for any two projects given in the list of sample projects or any other projects:

1. Development of problem statement.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Preparation of Software Configuration Management and Risk Management related documents.
4. Study and usage of any Design phase CASE tool.
5. Performing the Design by using any Design phase CASE tools.
6. Develop test cases for unit testing and integration testing
7. Develop test cases for various white box and black box testing techniques.

SAMPLE PROJECTS:

1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system

TEXT BOOKS:

1. Software Engineering, A Practitioner's Approach, Roger S. Pressman, 6th Edition, McGraw Hill International Edition
2. Software Engineering, Ian Sommerville, 7th Edition, Pearson Education
3. The Unified Modeling Language User Guide, Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. III Semester – CSE & IT	**L**	**T/P/D**	**C**
	0	**2**	**1**

(A19PC2IT02) PYTHON PROGRAMMING LABORATORY

COURSE OBJECTIVES:
- To install and run the Python interpreter
- To learn control structures
- To understand Lists, Dictionaries in python
- To handle Strings and Files in Python

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Develop the application specific codes using python
CO-2: Understand Strings, Lists, Tuples and Dictionaries in Python
CO-3: Verify programs using modular approach, file I/O, Python standard library
CO-4: Implement Digital Systems using Python

EXERCISE 1: Basics
Running instructions in Interactive interpreter and a Python Script
Write a program to purposefully raise Indentation Error and correct it

EXERCISE 2: Operations
Write a program to compute GCD of two numbers by taking input from the user
Write a program add.py that takes 2 numbers as command line arguments and prints its sum.

EXERCISE 3: Control Flow
Write a Program for checking whether the given number is even number or not.
Write a program using for loop that loops over a sequence.
Python Program to Print the Fibonacci sequence using while loop
Python program to print all prime numbers in a given interval (use break)

EXERCISE 4: Lists
Find mean, median, mode for the given set of numbers in a list.
Write a program to convert a list and tuple into arrays.
Write a program to find common values between two arrays.

EXERCISE 5: Dictionary
Write a program to count the numbers of characters in the string and store them in a dictionary data structure
Write a program combine_lists into a dictionary.

EXERCISE 6: Strings
Write a program to check whether a string starts with specified characters.
Write a program to check whether a string is palindrome or not

EXERCISE 7: Strings Continued
Python program to split and join a string
Python Program to Sort Words in Alphabetic Order

EXERCISE 8: Files
Write a program to print each line of a file in reverse order.
Write a program to compute the number of characters, words and lines in a file.
Write a program to count frequency of characters in a given file.

EXERCISE 9: Functions
Simple Calculator program by making use of functions
Find the factorial of a number using recursion
Write a function dups to find all duplicates in the list.
Write a function unique to find all the unique elements of a list.

EXERCISE 10: Functions - Problem Solving
Write a function cumulative_ product to compute cumulative product of a list of numbers.
Write a function reverse to print the given list in the reverse order.
Write function to compute GCD, LCM of two numbers

EXERCISE 11: Multi-D Lists
Write a program that defines a matrix and prints
Write a program to perform addition of two square matrices
Write a program to perform multiplication of two square matrices

EXERCISE 12: Modules
Install NumPypackage with pip and explore it.

EXERCISE 13:
Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR
Write a program to implement Half Adder, Full Adder, and Parallel Adder

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

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

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. III Semester	**L**	**T/P/D**	**C**
	0	**2**	**0**

(A19MN6HS02) ENVIRONMENTAL SCIENCE

COURSE PRE-REQUISITES: Basic knowledge of environmental issues

COURSE DESCRIPTION:
Environmental science is the study of patterns and processes in the natural world and their modification by human activity. We as human beings are not an entity, separate from the environment around us, rather we are a constituent seamlessly integrated and co-exist with the environment around us. To understand current environmental problems, we need to consider physical, biological and chemical processes that are often the basis of those problems. The course requires the students to identify and analyse natural and human- made environmental problems, evaluate the relative risks associated with these problems, and examine alternative solutions for resolving or preventing them. This course will survey some of the many environmental science topics at an introductory level, ultimately considering the sustainability of human activities on the planet. We are not an entity so separate from the environment that we can think of mastering and controlling it rather we must understand that each and every action of ours reflects on the environment and vice versa.

COURSE OBJECTIVES:
- To recognize the impacts of human interventions towards environment
- To list out the benefits in creating a sustainable environment
- To sketch out various activities in achieving a cleaner environment
- To emphasize the role of an individual for a better planet to live

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Gain a variety of experiences & acquire a basic knowledge about the environment & its allied problems
CO-2: Interpret the key components in safe guarding the environment
CO-3: Appraise the quality of environment in order to create a healthy atmosphere
CO-4: Familiarize with the individual responsibilities towards green revolution

MODULE 1: INTRODUCTION
Environmental Science: Introduction, Definition, scope and importance.

MODULE 2: AWARENESS ACTIVITIES
Small group meetings about:
- Water management
- Projects Vs Environment
- Generation of less waste
- Promotion of recycle use
- Impact of Science & Technology on Environment
- Avoiding electronic waste

MODULE 3: SLOGAN AND POSTER MAKING EVENT
- Food waste management

- Rain water harvesting
- Climate change
- Green Power
- Water conservation
- Green at work
- Role of IT in environment and human health
- Sustainable development

MODULE 4: EXPERT LECTURES ON ENVIRONMENTAL SCIENCE
- Environmental Impact Assessment
- Industrial waste treatment
- Organic farming/Vertical gardens/Hydroponics

MODULE 5: CLEANLINESS DRIVE
- Indoor air pollution
- Vehicular pollution
- VISUAL pollution
- Waste management at home
- Composting
- Plastic recycling

MODULE 6: CASE STUDIES
- HPCL disaster in Vizag
- Oleum gas leak in Delhi
- Mathura Refinery & Taj Mahal
- Conservation of Hussain Sagar lake
- The Cleanliest city of India-Surat
- Green Buildings in India
- KBR park in Hyderabad (Environmental protection Vs Development)
- Fluorosis
- Ecotourism & its impacts

TEXT BOOKS:
1. Environmental Studies for UG Courses, Erach Bharucha, UGC Publications, Delhi, 2004
2. Textbook of Environmental Studies, Deeksha Dave, S. S. Katewa, Cengage Delmar Learning India Pvt., 2012

REFERENCES:
1. Introduction to Environmental Science, Y. Anjaneyulu, BS Publications, 2004
2. Environmental Studies, Anubha Kaushik & C. P. Kaushik, 4th Edition, New Age International Publishers

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. IV Semester – CSE & IT	**L**	**T/P/D**	**C**
	3	**0**	**3**

(A19BS1MT11) PROBABILITY, STATISTICS AND QUEUING THEORY

COURSE PRE-REQUISITES: Permutations and Combinations, Basic Statistics

COURSE OBJECTIVES:
- To learn elementary ideas in basic probability
- To learn different types of probability distribution functions
- To learn methods of calculating correlation and regression
- To learn various methods to test the hypothesis
- To learn understand the basic concepts of queuing theory

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Solve problems involving basic probability
CO-2: Evaluate statistical parameters of different probability distributions
CO-3: Calculate correlation, regression, rank correlation coefficients
CO-4: Apply the knowledge of different probability distributions to Test of Hypothesis
CO-5: Apply the knowledge of different probability distributions to solve problems in queuing theory

UNIT – I:
Basic Probability: Sample space and events, Probability- The axioms of probability, some elementary theorems, conditional probability, Baye's theorem. Random variables - discrete and continuous distributions - Expectation of Discrete Random Variables, Moments, Variance of a sum.

UNIT – II:
Probability Distributions: Probability distributions: Binomial, Poisson and Normal - evaluation of statistical parameters for these three distributions –related properties. Box-Mueller Method, Transformation of Random Variables.

UNIT – III:
Testing of Hypothesis - Large Samples: Sampling distributions, Tests of hypothesis - null hypothesis, alternate hypothesis, type I, type II errors, critical region. Inferences concerning means and proportions- Large samples- test of hypothesis for single mean and difference between the means. Test of hypothesis for the proportions- single and difference between the proportions, confidence interval for the mean and proportions

UNIT – IV:
Tests of Significance - Small Samples: Tests of significance-t distributions, confidence interval for the t- distribution, F-distributions and Chi square distributions.

UNIT – V:
Correlation and Regression Analysis: Coefficient of correlation, Correlation Ratio, Logistic Regression, ANOVA Table, Multiple Regression model, Coefficient of Determination, Adjusted R2, Auto Correlation, Heteroskedasticity

UNIT – VI:

Queuing Theory: Queuing theory -Arrival process and service process- Pure birth and death process, M/M/1 model with finite and infinite capacities, M/M/C model with infinite capacity.

TEXT BOOKS:

1. Applied Probability, I. N. Blake, 9th Edition, John Wiley & Sons Inc., 1979
2. Introductory Statistics, Thomas H. Wonnacott & Ronald J. Wonnacot, John Wiley & Sons Inc., 1969
3. The Single Server Queue, J. W. Cohen, Wiley Interscience, 1969

REFERENCES:

1. Applied Statistics and Probability for Engineer, Douglas C. Montgomery, George C. Runger, 3rd Edition, John Wiley & Sons Inc., 2003
2. Probability and Statistics for Engineers, Richard A. Johanson, 5th Edition, Prentice-Hall, 1995
3. Applied Statistics for Engineers, Jay L. Devore, Nicholas R. Famum, Jimmy A. Doi, 3rd Edition, Cengage Learning
4. Some Problems in the Theory of Queues, D. G. Kendall, Journal of the Royal Statistical Society, Series B, 13, 151–185, 1951

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. IV Semester	**L**	**T/P/D**	**C**
	3	**0**	**3**

(A19HS1MG02) ENGINEERING ECONOMICS AND ACCOUNTANCY

(Common to EEE, EIE, CSE and IT)

COURSE OBJECTIVES:

- To explain the basic nature of pure economics and to analyse certain concepts of both Micro & Macro Economics and to know the role of managerial economics in solving problems of business enterprises
- To understand different forms of organizing private-sector and public-sector business enterprises and problems which have been encountered by public enterprises in India
- To describe each stage of product life cycle with the help of different costs and their role in maintaining optimum cost of production and overall profitability by considering different market competitions
- To analyse the process involved in preparation of project proposals, to estimate capital required to commerce and carryon business projects, to know the various sources of mobilizing required amount of capital and to evaluate investment opportunities
- To apply the basic accounting concepts & conventions and to analyse financial position of business enterprise

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Perform decision making function effectively in an uncertain framework by applying the concepts of economics, manage demand efficiently and plan future course of action

CO-2: Select suitable form of business organization which meets the requirements of business

CO-3: Fix the right price which can best meet the pre-determined objectives of the business under different market conditions

CO-4: Identify the best source of mobilising capital, select most profitable investment opportunity, carry out & evaluate benefit/cost, life cycle and Break-even analysis on one or more economic alternatives

CO-5: Analyze overall position of the business enterprise, therefore, take appropriate measures to improve the situation

UNIT – I:

Introduction to Economics & Managerial Economics: Introduction to Economics: Definition, nature, scope and types of Economics. Concepts of Macro-Economics: Gross Domestic Product (GDP), Gross National Product (GNP), National Income (NI) & Rate of Inflation.

Managerial Economics: Definition, nature, scope & significance.

Elements of Managerial Economics: Demand Analysis, Law of Demand, Elasticity of Demand and Demand Forecasting.

UNIT – II:

Forms of organizing Private and Public-Sector Business Enterprises:

Private Sector Business Enterprises:

(i) Sole Proprietorship - Definition, features, merits, limitations & suitability.
(ii) Partnership - Definition, Partnership Act, features, types, merits, limitations, suitability.
(iii) Joint-Stock Company - Definition, Companies Act, features, types, merits, limitations, suitability.
Public Sector Business Enterprises: Definition, features, objectives, merits, problems.

UNIT – III:
Market Structures, Product Life-Cycle (PLC), Pricing and Financial Accounting: Market Structures: Definition & common features of market and classifications of markets. Evaluation of market structures-Perfect Competition, Monopoly, Monopolistic Competition and Oligopoly.
Product Life-Cycle and Pricing: Definition, various stages of PLC, and Life-Cycle Costs; objectives and methods of pricing.
Introduction to Financial Accounting: Definition, basic principles and double-entry book- keeping, practice of accounting process-Journal, ledger, trial balance and final accounts (simple problems)

UNIT – IV:
Financial Analysis through Ratios: Meaning, computation of ratios
- **(i) Liquidity Ratios:** Current Ratio and Quick Ratio,
- **(ii) Solvency Ratios:** Interest Coverage Ratio and Debt- Equity Ratio,
- **(iii) Activity Ratios:** Stock/Inventory Turnover Ratio and Debt Turnover Ratio,
- **(iv) Profitability Ratios:** Gross Profit Ratio, Net Profit Ratio & Earning Per Share(EPS) Ratio.

UNIT – V:
Management Accounting: Definition & nature of Management Accounting.
Capital: Types of capital, factors influencing capital requirements, sources of mobilising Fixed and Working Capital.

UNIT – VI:
Cost Accounting: Cost Accounting: Definition, Types of costs – Opportunity cost, Explicit/Out-of-Pocket cost, Implicit/Imputed cost, Fixed cost, Variable cost, Semi-Variable cost, Differential cost, Sunk cost, Total cost, Average cost & Marginal cost. Break- Even/Cost-Volume-Profit (CVP) Analysis (Simple Problems).

TEXT BOOKS:

1. Managerial Economics and Financial Analysis, Aryasri, Tata McGraw-Hill, 2009
2. Managerial Economics, Varshney & Maheswari, Sultan Chand, 2009
3. Principles of Marketing: A South Asian Perspective, Kotler Philip, Gary Armstrong, Prafulla Y. Agnihotri and Eshan ul Haque, 13th Edition, Pearson Education/ Prentice Hall of India, 2010

REFERENCES:

1. Indian Economy, Misra S. K. and Puri, Himalaya Publishers
2. Textbook of Business Economics, Pareek Saroj, Sunrise Publishers
3. Financial Accounting for Management: An Analytical Perspective, Ambrish Gupta, Pearson Education
4. Managerial Economics, H. Craig Peterson & W. Cris Lewis, Prentice Hall of India
5. Guide to Proposal Writing, Jane C. Geever & Patricia McNeill, Foundation Centre

Website:
https://www.amazon.com/exec/obidos/tg/detail/-/0879547030/ref=ase_learnerassoci-20/102-4728473-7056968?v=glance&s=books

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. IV Semester – CSE & IT	**L**	**T/P/D**	**C**
	2	**1**	**3**

(A19PC1CS05) FORMAL LANGUAGES AND AUTOMATA THEORY

COURSE OBJECTIVES:

- To explain the theoretical foundations of computer science concerning– the relationships between languages and machines, the inherent limits of what can be computed, and the inherent efficiency of solving problems using machines such as FA, PDA, LBA and TM
- To identify a language's location in the Chomsky hierarchy (regular sets, context-free, context- sensitive, and recursively enumerable languages)
- To convert among equivalently powerful notations for a language, including among DFAs, NFAs, and regular expressions, and between PDAs and CFGs
- To build the foundation for students to pursue research in the areas of automata theory, formal languages, compiler design and computational power of machines

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: List computational devices according to their computational power, and tools which will allow us to tell if a device is powerful enough to solve a given computational problem

CO-2: Relate the concept of the grammar with the concept of programming language

CO-3: Design Solutions for problems related to Finite Automata, RE, CFG, PDA and Turing Machine

CO-4: Analyze various problems and categorize them into P, NP, NP-Complete and NP-Hard problems

UNIT – I:

Introduction: Alphabet, languages and grammars, Chomsky hierarchy of languages. Regular languages and finite automata: Deterministic Finite Automata (DFA), nondeterministic finite automata (NFA) and equivalence with DFA, NFA with ε - moves, Conversion to NFA without ε- moves, minimization of finite automata, equivalence between FAs, Finite Automata with Outputs – Mealy machine, Moore machine and equivalence.

UNIT – II:

Regular Languages and Finite Automata: Regular sets, Regular expressions and languages, Operations on Languages - Union, Concatenation, Kleen Closure, equivalence between finite automata and regular expressions, Regular grammars: Definition, productions, derivation, right linear and left linear grammars, and equivalence with Regular grammars and finite automata, properties of regular languages, pumping lemma for regular languages

UNIT – III:

Context-Free Languages: Context-Free Grammars (CFG) and Languages (CFL), parse trees, sentential forms, right most and left most derivations of strings, ambiguity in CFG, Left recursion and left factoring in context free grammars, Chomsky and Greibach normal forms, Pumping Lemma for context-free languages, closure properties of CFLs

UNIT – IV:

Pushdown Automata: definition, model, acceptance of CFL, Pushdown Automata (PDA), Acceptance by final state and acceptance by empty stack and its equivalence, Equivalence of CFG and PDA (proofs not required), Nondeterministic Pushdown Automata (NPDA), Context Sensitive Grammars: Context-Sensitive Grammars (CSG) and languages, Linear Bounded Automata (LBA) and equivalence with CSG.

UNIT – V:

Turing Machine: The basic model for Turing Machines (TM), Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, unrestricted grammars and equivalence with Turing Machines, nondeterministic TMs and equivalence with deterministic TMs, variants of Turing Machines.

UNIT – VI:

Computability Theory: Undecidability: Church-Turing Thesis, universal Turing Machine, undecidable problems about languages.LR (0) grammar, decidability of problems, Post's Correspondence Problem - The classes P and NP.

TEXT BOOKS:

1. Introduction to Automata Theory, Languages and Computations, H. E. Hopcroft, and J. D. Ullman, 2nd Edition, Pearson Education, 2003
2. Theory of Computer Science-Automata Languages and Computation, Mishra and Chandra Sekaran, 2nd Edition, PHI

REFERENCES:

1. Elements of the theory of Computation, H. R. Lewis and C. H. Papadimitriou, 2nd Edition, Pearson Education/PHI, 2003
2. Introduction to Languages and the Theory of Computation, J. Martin, 3rd Edition, TMH, 2003
3. Formal Languages and Automata Theory, K. V. N. Sunitha, N. Kalyani, 1st Edition, TMH, 2010
4. Automata and Computability, Dexter C. Kozen, Undergraduate Texts in Computer Science, Springer

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. IV Semester	**L**	**T/P/D**	**C**
	3	**0**	**3**

(A19PC1CS31) OPERATING SYSTEMS USING LINUX

COURSE OBJECTIVES:

- To understand the basic concepts and functions of operating systems
- To summarize various approaches to solve the problem of process concurrency in an operating system
- To evaluate the memory usage trade-offs in terms of size (main memory, auxiliary memory) and processor speed
- To understand disk storage strategies and file strategies with protection and security issues

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Identify system calls and evaluate process scheduling criteria of OS
CO-2: Develop procedures for process synchronization and scheduling services of an OS
CO-3: Analyze various mechanisms for disk access, file systems supported by an OS
CO-4: Elaborate the features of operating systems interns of virtual memory, protection and security aspects

UNIT – I:

Operating System Overview: Computer System hardware overview, Operating Structure functions and objectives, Evaluation of Operating system, Operating system services, System calls, System Programs, OS generation, System Boot, Operating System Structure, Basic Structure of Windows and Linux.
Processes and Threads: Introduction, operations on processes, Process Control Block (PCB), Process states and transitions, context switching, dispatcher. Threads, thread states, benefits of threads, types of threads, threading issues.

UNIT – II:

Process Scheduling: Types of schedulers, Scheduling Criteria, scheduling algorithms, multiprocessor, and real Time CPU scheduling.
Memory Management: Memory management techniques, fragmentation, paging, segmentation, paged segmentation

UNIT – III:

Inter-process Communication: Critical Section, race conditions, mutual exclusion, shared memory, message passing, semaphores and monitors, classical IPC Problems: producer-consumer, readers-writer and dinning philosopher
Deadlocks: Deadlock conditions, deadlock handling methods, RAG, Banker's algorithm, deadlock recovery

UNIT – IV:

Virtual Memory: Introduction, locality of reference, page fault, thrashing, working Set, demand paging, page replacement algorithms, allocation of frames.

File Management: File access methods, directory structure, file system structure, Allocation methods, directory implementation, efficiency, and performance.

UNIT – V:

Disk Management: Disk structure, Disk scheduling, reliability, disk formatting, swap space management

I/O: I/O devices, controllers, types of I/O, device drivers, Kernel I/O Structure, performance, Streams

UNIT – VI:

Linux System: Design principles, modules, Process management, scheduling, memory management, I/O management, file System, inter-process communication.

Mobile OS: iOS and Android – architecture and SDK framework, media layer, services layer, core OS layer, file system.

TEXT BOOKS:

1. Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 7th Edition, John Wiley
2. Unix Concepts and Applications, Sumitabha Das, 4th Edition, TMH, 2006

REFERENCES:

1. Modern Operating Systems, Andrew S Tanenbaum, 2nd Edition, Pearson/PHI
2. Operating Systems – A Concept Based Approach, D. M. Dhamdhere, 2nd Edition
3. Unix System Programming using C++, T. Chan, PHI
4. Operating Systems - Internal and Design Principles, Stallings, 5th Edition, Pearson Education/PHI, 2005

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. IV Semester – CSE & IT	**L**	**T/P/D**	**C**
	3	**0**	**3**

(A19PC1CS03) DESIGN AND ANALYSIS OF ALGORITHMS

COURSE OBJECTIVES:

- To reinforce algorithms analysis methods
- To analyse running time of an algorithm
- To understand different algorithm design strategies
- To familiarity with an assortment of important algorithms

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Apply algorithm design techniques and concepts to solve given engineering problem

CO-2: Analyze running times of algorithms using asymptotic analysis

CO-3: Develop efficient algorithms for computational tasks

CO-4: Computing complexity measures of algorithms

UNIT – I:

Introduction: Characteristics of algorithm. Analysis of algorithms: Asymptotic analysis of complexity bounds – best, average and worst-case behaviour; Performance measurements of Algorithm, Time and space trade-offs.

Divide and conquer General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication. Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem.

UNIT – II:

Fundamental Algorithmic Strategies: Greedy method: General method, applications-Job sequencing with dead lines, 0/1 knapsack problem, Minimum cost spanning trees, Single source shortest path problem, Huffman Codes.

UNIT – III:

Dynamic Programming: General method, Principle of optimality, applications-Multistage graphs, 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 coloring, Hamiltonian cycles.

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

UNIT – V:

Graph and Pattern Matching Algorithms: Graph Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS), connected and biconnected components, Topological Sorting.

Pattern Matching Algorithms: Brute Force method, Knuth-Morris-Pratt algorithms

UNIT – VI:

NP Hard and NP-Complete problems: P, NP, NP-complete and NP-hard. Cook's theorem

Randomized Algorithm: Hiring Problem, Randomized Quick Sort

TEXT BOOKS:

1. Fundamentals of Computer Algorithms, E. Horowitz et al., Galgotia Publications
2. Introduction to Algorithms, Thomas H. Cormen, Charles E. Lieserson, Ronald L. Rivest and Clifford Stein, 4th Edition, MIT Press/McGraw-Hill

REFERENCES:

1. Algorithm Design, Jon Kleinberg and Eva Tardos, 1st Edition, Pearson
2. Algorithm Design: Foundations, Analysis, and Internet Examples, Michael T. Goodrich and Roberto Tamassia, 2nd Edition, Wiley
3. Algorithms – A Creative Approach, Udi Manber, 3rd Edition, Addison-Wesley, Reading, MA
4. Introduction to the Design and Analysis of Algorithms, Anany Levitin, 3rd Edition, Pearson Publications

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. IV Semester – CSE & IT	**L**	**T/P/D**	**C**
	3	**0**	**3**

(A19PC1CS04) DATABASE MANAGEMENT SYSTEMS

COURSE OBJECTIVES:

- To introduce the concepts of Data Base Management and to give the description of structure of Data Base systems
- To understand concepts of ER model and model the data base for the given scenarios and prepare the database through normalization
- To know the features of various models of data and query representations
- To introduce the concepts and protocols related to transaction management and understand the concepts of data storage

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Appreciate the underlying concepts of database system architecture and technologies
CO-2: Develop database schema for a given scenario
CO-3: Query the database using the relevant programming language
CO-4: Design schedules using multiple transactions

UNIT – I:
Introduction to Databases and Database Management System: Database system Applications, Advantages of DBMS over File System, Data Models, Instances and schema, View of Data, Database Languages –DDL, DML, DCL, Database Users and Administrator, Database System Architecture.

UNIT – II:
Database Design and ER Diagrams: Attributes and Entity Sets, Relationships and Relationship Sets, Constraints, Keys, Design Issues, Entity-Relationship Diagram, Extended E-R Features, Database Design with ER model, Database Design for a schema.

UNIT – III:
Introduction to the Relational Model: Structure of RDBMS, Integrity Constraints over Relations, Querying Relational Data, Relational Algebra and Relational Calculus.
Introduction to SQL: Data Definition commands, Data Manipulation Commands, Basic Structure, Set operations Aggregate Operations, Join Operations, Sub queries and correlated queries, views, Triggers, Cursors, Embedded SQL, Overview of NoSQL database.

UNIT – IV:
Functional Dependencies: Introduction, Basic Definitions, Trivial and Non trivial dependencies, closure of a set of dependencies, closure of attributes, irreducible set of dependencies.
Schema Refinement in Database Design: Problems Caused by Redundancy, Decompositions – Problem Related to Decomposition, Lossless Join Decomposition, Dependency Preserving Decomposition, FIRST, SECOND, THIRD Normal Forms, BCNF, Multivalued Dependencies, FOURTH Normal Form.

UNIT – V:

Transaction Management: Transaction state, Implementation of atomicity and Durability, Concurrent executions – Serializability, Recoverability.

Concurrency Control: Lock Based Protocols, Timestamp Based Protocols, Validation Based Protocols, Multiple Granularity, Dead Lock Handling

Recoverability: Failure Classification, Storage Structure, Recovery and Atomicity- Log Based recovery, Recovery with concurrent transactions, Checkpoints.

UNIT – VI:

File Organization: Organization of records in file, Data Dictionary Storage.

Indexing and Hashing: Basic Concepts, Ordered Indices+ Tree Index files, B tree index files – Static Hashing – Dynamic Hashing – Comparison of Indexing with Hashing.

TEXT BOOKS:

1. Database System Concepts, Silberschatz, Korth, Sudarshan, 7th Edition, McGraw-Hill
2. Introduction to Database Systems, C. J. Date, Pearson Education

REFERENCES:

1. Database Management Systems, Raghu Ramakrishnan, Johannes Gehrke, Tata McGraw-Hill
2. Fundamentals of Database Systems, Elmasri Navathe Pearson Education
3. Database Systems Design, Implementation, and Management, Peter Rob & Carlos Coronel, 7th Edition, Cengage Learning

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. IV Semester	**L**	**T/P/D**	**C**
	0	**3**	**1.5**

(A19PC2CS31) OPERATING SYSTEMS USING LINUX LABORATORY

COURSE OBJECTIVES:

- To apply the basic commands to handle the Linux Environment
- To understand the Shell/C scripting constructs to modify the file system content
- To explore scheduling, synchronization and deadlock mitigation techniques
- To analyze various file, disk, and memory management mechanisms

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Implement various Shell Scripts in managing Linux Environment
CO-2: Construct C Scripts to handle the disk access, file systems facilities in Linux
CO-3: Implement the IPC Mechanisms in Linux operating system using C language
CO-4: Design and implement various disk management, processor and memory scheduling algorithms

EXPERIMENTS:

WEEK 1-3:

1. Explore basic Linux utilities and vi editor features.
2. Implementation of Shell Programming (I/O, Decision making, Looping, Multi-level branching)
3. Creating child process using fork() system call, Orphan and Zombie process creation and Terminating a process, Demonstration of process management and thread management system calls

WEEK 4-5:

4. Simulate CPU scheduling algorithms
   a) FCFS b) SJF c) Priority d) Round Robin
5. Simulate the following Memory Allocation Methods for fixed partition
   a) First Fit b) Worst Fit c) Best Fit

WEEK 6-7:

6. Implementation of Paging Technique of Memory Management
7. Demonstration of IPC mechanisms shared memory and message passing in linux

WEEK 8

8. Implementation of producer-consumer, readers-writer and dining philosopher's problem using semaphores
9. Implementation of Bankers Algorithm for Deadlock Avoidance

WEEK 9-10:

10. Simulate the following Page Replacement Algorithms
    a) FIFO b) LRU c) LFU

WEEK 11-12:

11. Simulate the following File Allocation Strategies

a) Sequential b) Indexed c) Linked

WEEK 13-14:

12. Simulate the following Disk scheduling algorithms
    a) FCFS b) SSTF c) Scan d) look

TEXT BOOKS:

1. Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 7th Edition, John Wiley
2. UNIX Concepts and Applications, Sumitabha Das, 4th Edition, TMH, 2006

REFERENCES:

1. Modern Operating Systems, Andrew S. Tanenbaum, 2nd Edition, Pearson/PHI
2. Operating Systems – A Concept Based Approach, D. M. Dhamdhere, 2nd Edition
3. Unix System Programming using C++, T. Chan, PHI
4. Operating Systems - Internal and Design Principles, Stallings, 5th Edition, Pearson Education/PHI, 2005

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B.Tech. IV Semester – CSE & IT	**L**	**T/P/D**	**C**
	0	**3**	**1.5**

(A19PC2CS01) DATABASE MANAGEMENT SYSTEMS LABORATORY

COURSE OBJECTIVES:
- To provide the fundamental concepts of database creation
- To implement the concepts of Data manipulation
- To develop procedures for querying Multiple tables
- To understand the concepts of PL / SQL

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Apply Integrity constraints for creating consistent RDBMS environment
CO-2: Implement SQL functions using the DUAL table
CO-3: Create, maintain and Manipulate the Data through SQL commands
CO-4: Develop Triggers, query through PL /SQL structures

WEEK 1:

Implement the following using DUAL table:
a) Character functions,
b) Numeric functions
c) Date functions and
d) Conversion functions.

WEEK 2:

Practice DDL and DML commands on a basic table without integrity constraints.

WEEK 3:

Practice DDL and DML commands on a Relational Database, specifying the Integrity constraints.
(Primary Key, Foreign Key, CHECK, NOT NULL)

WEEK 4:

Apply the concepts of Joins, SET operations and SQL functions on any two relational schemas

WEEK 5-7:

Apply the concepts of Joins, SET operations and SQL functions on the following schema:

a) Employee:

Name	**Datatype**	**width**	**Constraint**	**Description**
Empno	Integer	4	Primary Key	Employee Number
Ename	Varchar	20		Employee Name
Job	Char	12		Designation
Mgr	Integer	4		Manager Number

Hiredate	Date			
Sal	Number	(8,2)		Salary
Comm	Number	(6,2)		Commission
Deptno	Integer	2	Foreign Key	Department Number

b) Dept:

Name	Datatype	width	Constraint	Description
Deptno	Integer	2	Primary Key	Department Number
Dname	Varchar	12		Department Name
Loc	Char	10		Location

c) Salgrade:

Name	Datatype	width	Constraint	Description
Grade	Integer	1		Grade
Hisal	Integer	4		Upper scale of salary
Losal	Integer	5		Lower scale of salary

WEEK 8 – 11:

End to end implementation of a schema for a specific system along with the illustrations of querying.

A system is described by specifying the functional and non-functional requirements. Based on this description, the major entities are identified and modelled. Further the relationships are modelled to form the initial schema. The schema is further refined by removing redundancies through normalization. Also based on the query requirements, the schema is remodelled to facilitate querying. Finally an illustration of various queries to extract required information from the system is shown using SQL / MYSQL.

The five major workflows to be implemented are:

1. System Specification
2. Design of Initial Schema
3. Schema refinement using functional dependencies and normalization
4. Schema refinement using query requirements
5. Illustration of querying the system using SQL / MYSQL.

WEEK 12:

Implementation of PL / SQL concepts

WEEK 13:

Creating and executing CURSORS.

WEEK 14:

Creation and application of TRIGGERS on a Relational schema.