GOVT. COLLEGE OFENGINEERING,
AMRAVATI

B. TECH. (Information Technology)
FOURTH SEMESTER
CURRICULUM
Department of Information Technology

GOVERNMENT COLLEGE OFENGINEERING, AMRAVTI.

Department of Information Technology. Proposed

Scheme for B. Tech. (Information Technology)

SEM IV

Category	Course Code	Nameofthe Course	Teaching Scheme				Evaluation Scheme						Credits
							Theory			Practical			
			Theory Hrs /week	Tutorial Hrs/week	Practical Hrs/week	Total	MSE	TA	ESE	ICA	ESE		
PCC	ITU421	Discrete Mathematics	3	1	--	4	30	10	60	---	--	100	4
PCC	ITU422	Database Management Systems	3	--	--	3	30	10	60	---	--	100	3
PCC	ITU423	Operating System	3	--	--	3	30	10	60	---	--	100	3
PCC	ITU424	Design & Analysis of Algorithms	3	--	--	3	30	10	60	---	--	100	3
HSMC	ITU425	Organizational Behavior	3	--	--	3	30	10	60	---	--	100	3
MC	SHU422	Environmental Sciences	-	--	--	--	--	--	60	---	-	60	-
PCC-LC	ITU426	Database Management Systems Lab	---	--	2	2	---	----	---	25	25	50	1
PCC-LC	ITU427	Operating System Lab	---	--	2	2	---	--- --	---	25	25	50	1
PCC-LC	ITU428	Design & Analysis of Algorithms Lab	---	--	2	2	---	--	---	25	25	50	1
PCC-LC	ITU429	Python Programming Lab	---	--	4	4	---	--	---	25	25	50	2
		Total	15	1	10	26	150	---50	360	100	100	760	21

TA : Teacher Assessment **CT: Class Tests** **ESE: End Semester Examination** **ICA :Internal Continuous Assessment**

Government College of Engineering, Amravati

Department of Information Technology

Program Educational Objectives

PEO 1: To formulate, analyze and solve real life problems in software industry, research academia and society at large.

PEO 2: To provide opportunity to learn the latest trends in information technology and prepare for lifelong learning process.

PEO 3: To exhibit strong communication and interpersonal skills, broad knowledge, and global perspectives to work effectively and ethically in multidisciplinary teams.

Program Specific Outcomes

PSO 1: To develop technically sound human resource that shows inclination to pursue IT career in profession, research and higher education.

PSO 2: To exhibit the knowledge of algorithms, data structures /management, software design, information security, programming languages, computer organization and architecture and data science and analytics as a IT professional.

ITU421 DISCRETE MATHEMATICS

Teaching Scheme : 03 L+ 01 T Total 04 Credits : 04
Evaluation Scheme: 30 MSE+10 TA+ 60 ESE Total Marks: 100
Duration of ESE : 2Hrs.30min

Course Objectives

I. Introduce students to the techniques, algorithms, and reasoning processes involved in the study of discrete mathematical structures.
II. Introduce students to set theory, inductive reasoning, elementary and advanced counting techniques, equivalence relations, recurrence relations, graphs, and trees.
III. Express a logic sentence in terms of predicates, quantifiers, and logical connectives.
IV. Apply rules of inference, tests for validity, and methods of proof including direct and indirect proof forms, proof by contradiction, proof by cases, and mathematical induction and write proofs using symbolic logic and Boolean Algebra.
V. Determine if a given graph is simple or a multigraph, directed or undirected, cyclic or acyclic, and determine the connectivity of a graph.

Set Theory , Logic and Proofs : Propositions, Conditional Propositions, Logical Connectivity, Propositional calculus, First order logic, Proofs: Proof Techniques, Mathematical Induction. Set, Combination of sets, Principle of inclusion and exclusion , strong Induction .

Relations, Functions, Recurrence Relations: Definitions, Properties of Binary Relations, Equivalence Relations and partitions, Partial ordering relations and lattices, Chains and Anti chains. Theorem on chain, Recurrence relations. Functions: Definition, Domain, Range, Image, etc. Types of functions: Surjection, Injection, Bijection, Inverse, Identity, Composition of Functions.

Number Theory: Basics of Modulo Arithmetic, Basic Prime Number Theory, GCD, LCM, Divisibility, Euclid's algorithm, Factorization, Chinese Remainder Theorem Fields: Naturals, Integers, Rationals, Reals, Complex Numbers Properties of operations: associative, commutative, distributive, identity, inverse .

Counting Basic Counting Techniques (sum, product, subtraction, division, exponent): Pigeonhole and Generalized Pigeonhole Principle with many examples, Permutations and Combinations and numerical problems,Identity and Triangle, Generating Permutations and Combinations

Graphs & Trees Basic terminology:multi graphs and weighted graphs, paths and circuits, shortest path Problems, Euler and Hamiltonian paths and circuits, factors of a graph, planar graph, independent sets, graph coloring. Trees, rooted trees, path length in rooted trees, binary search trees, spanning trees and cut set, theorems on spanning trees, cut sets , circuits, minimal spanning trees, Kruskal’s and Prim’s algorithms for minimal spanning tree.

Algebraic Systems: Algebraic Systems, Groups, Semi Groups, Monoids, Subgroups, Permutation Groups, Codes and Group codes, Isomorphism and Automorphisms, Homomorphism and Normal Subgroups, Ring, Field.

Text Books

1. C. L. LIU, "Elements of Discrete Mathematics", 2nd Edition, Tata McGraw-Hill, 2002, ISBN: 0-07- 043476-X.
2. G. Shanker Rao, "Discrete Mathematical Structures", New Age International, 2002, ISBN: 81-224- 1424-9 Reference Books:
3. Lipschutz, Lipson, Discrete Mathematics, 2nd Edition, Tata McGraw-Hill, 1999, ISBN 007-463710--X.
4. V. K. Balakrishnan, Graph Theory, TMH (Recommended for Graph), ISBN 0-07-058718-3
5. B. Kolman, R. Busby and S. Ross, "Discrete Mathematical Structures", 4th Edition, Pearson Education, 2002, ISBN 81-7808-556-9
6. J. Tremblay, R. Manohar, "Discrete Mathematical Structures with application to Computer Science", McGraw-Hill, 2002 ISBN 0-07-065142-6 (Recommended for prepositional Calculus).
7. Kenneth H. Rosen: Discrete Mathematics and Its Applications, 5th Edition, Tata McGrawHill, 2003, ISBN 0-07-053047-5. **Course Outcomes**

Students will be able to:

ITU 421.1	Explain basic terminology, formal logic, proofs, sets, relations, functions, recursion
ITU 421.2	Use formal logic proof and logical reasoning to solve problems
ITU 421.3	Relate the ideas of mathematical induction to recursion and recursively defined structures
ITU 421.4	Solve problems based on graphs, trees and related algorithms
ITU 421.5	Relate, interpret and apply the concepts to various areas of computer science

ITU422 DATABASE MANAGEMENT SYSTEMS

Teaching Scheme : 03 L **Total 03** **Credits : 03**

Evaluation Scheme: 30 MSE +10 TA+ 60 ESE **Total Marks: 100**

Duration of ESE: 2Hrs.30min

Course Objectives

I. Analyze database requirements and determine the entities involved in the system and their relationship to one another.
II. Design ER-models to represent simple database application scenarios
III. Devise queries using Relational Algebra, Relational Calculus and SQL
IV. To be familiar with basic database storage structures
V. Develop an understanding of essential DBMS concepts such as: database integrity, concurrency

Database system architecture: Introduction to database management system, Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML). **Data models:** Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations.

Relational query languages: Relational algebra, tuple and domain relational calculus, SQL and QBE.

Relational database design: Domain and data dependency, Armstrong's axioms, Normal forms (1NF,2NF,3NF), Dependency preservation, Lossless design.

Query processing and optimization: Forms of a basic SQL query Evaluation of relational algebra expressions, Query equivalence, Join strategies.

Storage structures: Indices, B-trees, hashing.

Transaction processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.

Text books

1. "Database System Concepts", 5th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.
2. "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education

Reference books

1. "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
2. "Database Management Systems", Raghu Ramakrishnan, Mcgraw-Hill Education

3. “Fundamentals of Database Systems”, By: Elmasri and Navathe, 4th Edition Practical PostgreSQL O’REILLY

Useful link:

https://nptel.ac.in/courses/106105175/, IIT Kharagpur

https://nptel.ac.in/courses/106106093/, IIT Madras

https://nptel.ac.in/content/storage2/courses/106106095/pdf/1_Introduction.pdf

Course outcomes

ITU422.1 Design E-R Model for given requirements and convert the same into database tables and normalization.

ITU422.2 Create databases in an RDBMS and enforce data integrity constraints using SQL.

ITU422.3 Solve real world problems using appropriate set, function, and relational models.

ITU422.4 Understand the principles of storage structure and recovery management.

ITU422.5 For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.

ITU422.6 Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling

ITU 423 OPERATING SYSTEM CONCEPTS

Teaching Scheme: 03 L+00T **Total:03** **Credits: 03**
Evaluation Scheme: 30 MSE+10 TA+ 60 ESE **Total Marks: 100**
Duration of ESE: 2hrs.30min.

Course Objectives

I. To learn the mechanisms of OS to handle processes and threads and their communication. Perform operations on processes and threads.

II. To learn the mechanisms involved in memory management in contemporary OS.

III. To gain knowledge on distributed operating system concepts that includes Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols.

IV. To know the components and management aspects of file management. V. To learn to implement simple OS mechanisms.

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS-Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and WINDOWS Operating System.

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads, Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Preemptive and Non pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.

Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson’s Solution, The Producer\ Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader’s & Writer Problem, Dinning Philosopher Problem etc.

Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker’s algorithm, Deadlock detection and Recovery.

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging. Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault , Working Set , Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).

I/O Hardware: I/O devices, Device controllers, Direct memory access, Disk structure, Disk scheduling algorithms File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN.

Text books

1. Operating System Concepts Essentials, 9th Edition by Avi Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.

Reference books

1. Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing.
2. Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, AddisonWesley.

3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India.

4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates.

Course Outcomes

ITU 423.1 Create processes and threads.

ITU 423.2 Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.

ITU 423.3 For a given specification of memory organization develop the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time.

ITU 423.4 Simulate file management system.

ITU 423.5 For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.

ITU424 DESIGN AND ANALYSIS OF ALGORITHMS

Teaching Scheme : 03 L Total: 03 Credits : 03
Evaluation Scheme: 30 MSE +10 TA+ 60 ESE Total Marks: 100
Duration of ESE : 2Hrs.30min

Course Objectives

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

Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters’ theorem.

Divide and Conquer Technique

General Method, Revision and analysis of merge sort, quick sort and binary search, counting inversions, Finding closest pair of points, Integer multiplication.

Greedy Method

Elements of greedy technique, Activity selection problem, Fractional Knapsack Problem, Job Sequencing problem, Huffman Coding, Finding Single Source Shortest path in graph:
Dijkstra's Algorithm, Revision and analysis of minimum spanning tree algorithms.

Dynamic Programming

Elements of Dynamic Programming, Principles of Dynamic programming- memorization or iteration over sub problems, Assembly line scheduling, Matrix chain multiplication, Longest common subsequence, All pair shortest path algorithm- Floyd-Warshall's Algorithm..

NP-Completeness

Matching , Introduction to NP-Complete , Search/Decision, SAT, Independent Set, 3VC ,Exact Cover, Multi Set, Subset Sum & Partition, Hamiltonian Circuit.

Approximation Algorithms

The vertex-cover problem, The set-covering problem, The subset sum problem. **Text Books**

1 T. H. Cormen, C. E. Leiserson, R. L.Rivest and C. Stein, “Introduction to Algorithms”, MIT Press/McGraw Hill, Second Edition.

2 Jon Kleinberg, Eva Tardos, “Algorithm Design”, Pearson, Addison Wesley

Reference books

1 V. Aho, J. E Hopecroft and J.D. Ullman, The design and analysis of algorithm, Addision-Wesley, 1974

Course Outcomes

ITU 424.1 For a given algorithms analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms.

ITU 424.2 Describe the greedy paradigm and explain when an algorithmic design situation calls for it. For a given problem develop the greedy algorithms.

ITU 424.3 Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Synthesize divide-and-conquer algorithms. Derive and solve recurrence relation.

ITU 424.4 Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. For a given problems of dynamic-programming and develop the dynamic programming algorithms, and analyze it to determine its computational complexity.

ITU 424.5 Student will develop ability to identify weather given problem is NP-Complete or not, and develop efficient algorithm that gives good solution.

ITU425 ORGANIZATIONAL BEHAVIOUR

Teaching Scheme:	**03 L + 00T**	**Total: 03**	**Credits : 03**
Evaluation Scheme:	**30 MSE +10 TA+ 60 ESE**		**Total Marks: 100**
Duration of ESE:	**02 Hrs. 30 min**		

Course Objectives

To expose the students to the following:

I. To familiarize students with the principles of business management and organizational behaviour in general.

II. To understand individual and group actions in the workplace in particular in order to increase an organization's effectiveness.

III. The course will use Indian and global organizational perspectives, strategies and range of other case studies and examples to have a global view of the subject.

IV. To imbibe a correct sense of organization and roles to be played during career.

V. To inculcate a sense of cognitive and behavioral understanding of the world as a workplace.

Meaning and concept of organizational behaviour: Theories of the Organization; Personality: meaning, factors affecting personality, Big five model of personality; Learning: concept and theories of learning (Classical conditioning, operant conditioning and social learning theory), concept of reinforcement; Perception: concept, factors affecting perception, process of perception, perceptual errors.

Motivation: Concept, importance, Content theories; Maslows Need Theory, Alderfers ERG theory (Existence, Relatedness and Growth), McCllelands Theory of Needs, Herzbergs DualFactor Theory and Process theories; Adams Equity Theory, Vrooms Expectancy Theory **Leadership:** Concept, Theories (Trait, Behavioural, Contingency, Charismatic, Transactional and

Transformational Leadership; Emotional Intelligence: Concept, Importance, Dimensions. **Teams and Groups:** Definition, Team Structure and Effectiveness, Stages of Team/Group Development, Team/Group Cohesiveness;Goal-Setting, Beyond Self-Interest, Analysis of Interpersonal Relationship: Transactional Analysis, Conflict: Concept, Sources, Types, Stages of Conflict, Management of Conflict.

Organizational Power: Sources of Power and Dysfunctional uses of Power; Organizational Change: Concept, Resistance to change, Managing resistance to change, Kurt Lewin Theory of Change; Organizational Development (OD): Meaning and types of OD Interventions.

Text Books

1. Robbins, Stephen P and Judge T.A. (2013). Organisational Behaviour,15th Edition, Pearson.
2. Stephen, P. Robbins and Mary, Coulter (2010). Management ,9th Edition, Pearson
3. Kaul, Vijay Kumar (2012). Business Organisation and Management - Text and Cases. Pearson
4. Singh, Kavita. Organisational Behaviour,3rd Edition,Vikas Publication.

Reference Books

1. De Cenzo, D.A. and Robbins.(2010). Fundamentals of Human Resource Management (10th Edition). New York: John Wiley and Sons
2. ArunMonappa and Miza S Saiyadain (1999).Personnel Management (2 nd Edition). New Delhi: Tata McGraw-Hill publishing Company Limited

Course Outcomes

ITU 425.1 Understand the dynamics of organizational behaviour, and explain management roles with a comprehensive view of organizational behaviour.

ITU 425.2 Knowing the specific aspects of contemporary organizational behavior.

ITU 425.3 Gain an appreciation of the different approaches to organizational structures.

ITU 425.4 Understand personality, learning and emotional function at work along with team formation and working.

ITU 425.5 Comprehending the concept of motivation, leadership, power and conflict and team building.

ITU 425.6 Understand the fundamentals of group actions and the organizational change and growth process.

ITU426 DATABASE MANAGEMENT SYSTEMS LAB

Teaching Scheme : 02 P **Total 02** **Credit : 01**
Evaluation Scheme: 25 ICA+25ESE **Total Marks: 50**
Duration of ESE : 3Hrs.

Course Objectives

I. Develop the logical design of the database using data modeling concepts such as entityrelationship diagrams.
II. Be able to understandbasic database concepts, applications, data models, schemas and instances.
III. To demonstrate the use of constraints and utilize a wide range of features available in a DBMS package.
IV. Manipulate a database using SQL.

Suggested List of Experiments but not limited to the given list of experiments:

Students have to perform minimum 10 practical's, which should include one mini project.

1. Student should decide on a case study and formulate the problem statement.
2. Conceptual designing using ER diagrams (identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.
3. To execute and verify the data definition language commands
    1. create
    2. alter
    3. drop
    4. truncate
    5. comment
    6. rename
4. Create table employee(empno number(4) primary key, ename varchar2(10), job varchar2(6), sal number(5),deptno number(7));operate following queries on employee table:
    1. write a query to add a new column in to employee

    2. write a query to add multiple columns in to employee

5. To execute and verify the dml and tcl language commands dml (data manipulation language)
    1. select
    2. insert
    3. delete
    4. update
6. Normalization -to remove the redundancies and anomalies in the above relational tables, normalize up to third normal form.
7. To execute and verify the dcl language commands tcl (transaction control language)
    1. commit
    2. roll back
    3. Savepoint
8. To execute and verify the sql commands for nested queries.
9. To execute and verify the sql commands using join queries.
10. To execute and verify the sql commands for views.
11. To write a pl/sql block using different control (if, if else, for loop, while loop.) statements.
12. Mini Project

Course Outcomes

At the end of the course the students are able to:

ITU426.1 Apply the basic concepts of Database Systems and Applications.

ITU426.2 Use the basics of SQL and construct queries using SQL in database creation and interaction.

ITU426.3 Design a commercial relational database system (Oracle, MySQL) by writing SQL using the system.

ITU426.4 Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.

ITU426.5 Formulate, using SQL, solutions to a broad range of query and data update problems.

ITU427 OPERATING SYSTEM CONCEPTS LAB

Teaching Scheme : 02 P **Total 02** **Credits : 01**
Evaluation Scheme: 25 ICA +25 ESE **Total Marks: 50**
Duration of ESE: 3Hrs.

Course Objectives

I. To familiarize the students with the Operating System.
II. To demonstrate the process, memory, file and directory management issues under the LINUX operating system.
III. To introduce LINUX basic commands for operating system concepts.
IV. To make students how to make simple programs in LINUX and administrative task of LINUX.

List of Experiments

1. Basic Concepts : Introduction to Linux Operating System , basic commands in Linux and Writing shell scripts in Vi Editor.
2. Process Management:

   i)Use **ps** to search for the **init** process by name. ii) What is the **process id** of the **init** process? iii) Use the **who am i** command to determine your terminal name.

   iv) Using your terminal name from above, use **ps** to find all processes associated with your terminal.

   v) What is the **process id** of your shell?

   vi) What is the **parent process id** of your shell ?

   vii) Start two instances of the **sleep 3342** in background. viii) Locate the **process id** of all **sleep** commands. ix)Display only those two **sleep** processes in **top**. Then quit top.

   x)Use a **standard kill** to kill one of the **sleep** processes.

3. Process Priorities

   i). Create a new directory and create six **pipes** in that directory.

   ii). Bounce a character between two **pipes**.

   iii). Use **top** and **ps** to display information (pid, ppid, priority, nice value, ...) about these two cat processes.

   iv). Bounce another character between two other pipes, but this time start the commands **nice**. Verify that all **cat** processes are battling for the cpu. (Feel free to fire up two more cats with the remaining pipes).

V. Use **ps** to verify that the two new **cat** processes have a **nice** value. Use the -o and -C options of **ps** for this.

vi). Use **renice** te increase the nice value from 10 to 15. Notice the difference with the usual commands.

4. Disk partitions

i). Use **fdisk -l** to display existing partitions and sizes. ii). Use **df -h** to display existing partitions and sizes.

iii). Compare the output of **fdisk** and **df**. iv). Create a 200MB primary partition on a small disk.

v). Create a 400MB primary partition and two 300MB logical drives on a big disk.

vi). Use **df -h** and **fdisk -l** to verify your work.

vii). Compare the output again of **fdisk** and **df**. Do both commands display the new partitions? viii). Create a backup with **dd** of the **mbr** that contains your 200MB primary partition. ix). Take a backup of the **partition table** containing your 400MB primary and 300MB logical drives. Make sure the logical drives are in the backup.

x). Remove all your partitions with fdisk. Then restore your backups.

5. Logical Volume Management:

i). Create a volume group that contains a complete disk and a partition on another disk. ii). Create two logical volumes (a small one and a bigger one) in this volume group. Format them with ext3, mount them and copy some files to them.

6. File systems

i). List the file systems that are known by your system. ii).Create an **ext2** file system on the 200MB partition. iii).Create an **ext3** file system on one of the 300MB logical drives.

iv).Create an **ext4** on the 400MB partition.

v).Set the reserved space for root on the ext3 file system to 0 percent.

vi).Verify your work with **fdisk** and **df**.

vii).Perform a file system check on all the new file systems.

7. Scheduling

i). Schedule two jobs with **at**, display the **at queue** and remove a job.

ii). As normal user, use **crontab -e** to schedule a script to run every four minutes.

iii). As root, display the **crontab** file of your normal user. iv). As the normal user again, remove your **crontab** file.

v). Take a look at the **cron** files and directories in **/etc** and understand them. What is the **runparts** command doing ?

8. Memory Management

i). Use **dmesg** to find the total amount of memory in your computer. ii). Use **free** to display memory usage in kilobytes (then in megabytes).

iii). On a virtual machine, create a swap partition (you might need an extra virtual disk for this). iv). Add a 20 megabyte swap file to the system.

v). Put all swap spaces in **/etc/fstab** and activate them. Test with a reboot that they are mounted.

vi). Use **free** to verify usage of current swap.

Course outcomes

Upon the completion of Operating Systems Concepts practical course, the student will be able to:

ITU427.1 Apply basic commands in Linux for understanding OS concepts.

ITU427.2 Recognize CPU Scheduling, synchronization, and deadlock.

ITU427.3 Use Linux commands, and develop various system programs under Linux to make use of OS concepts related to process synchronization, shared memory, file systems.

ITU428 DESIGN AND ANALYSIS OF ALGORITHMS LAB

Teaching Scheme : 02 P	**Total 02**	**Credits : 01**
Evaluation Scheme: 25 ICA +25 ESE		**Total Marks: 50**
Duration of ESE: 3Hrs.		

Course Objectives

I. Analyze Algorithm depending upon its time complexity & Space complexity

II. Study of various types of Algorithms :-

a. Greedy Algorithm,

b. Divide & conquer Algorithm,

c. Dynamic Programming.

III. Identifying the type of problems NP, NP hard

List of Experiments:-

1.Apply Heap sort technique on a given set of elements.

2.Develop a simulator for a given set of elements using Merge sort technique / Selection sort technique .

3. Develop a simulator for a given set of elements using Quicksort technique .

4. Check whether a graph is connected using Depth first Search technique.

5.Implement 0/1 knapsack problem using greedy method.

6. Find shortest paths to other vertices using Dijkstra's algorithm.

7. A minimum cost spanning tree for a given undirected graph using Prim's algorithm or Kruskal's algorithm.

8. Print all the nodes reachable from a given starting node in a digraph using Breadth first search technique.

9. Develop a simulator for a pair shortest paths problem using Floyd's algorithm.

10. Design a simulator for n-Queens problem using backtracking technique.

Course outcomes:

ITU 428.1 Ability to write programs to solve problems using algorithm design techniques such as Divide and Conquer.

ITU 428.2 Ability to write programs to solve problems using algorithm design techniques such as Greedy.

ITU 428.3 Ability to write programs to solve problems using algorithm design techniques such as Dynamic programming.

ITU 429 PYTHON PROGRAMMING LAB

Teaching Scheme: 04 P Total: 04 Credit: 02
Evaluation Scheme: 25 ICA+ 25 ESE Total Marks: 50
Duration of ESE: 3 Hrs.

Course Objective

I Exposing students to free open source software, Python and to open source packages freely available.

II Enabling students to learn many open source aspects of python and help them learn to implement these to variety of problems including real-life problems and solution finding process

III Introduce open source software paradigm to the students to review popular open source software licenses, the structure of an open source project, establishing a collaborative team and software creation and current open source world events.

IV Motivation to learn variety of programming languages to be able to compete with peers around the world and participate in world programming events.

Suggested List of Experiments/Assignments but not limited to the given list of experiments:
(Note: a. Experiments/assignments can be given to students by the instructor as per current scenario of workability and availability of the technology with a flexibility of students' choice in selecting the experiments from the given list. b. Experiments aim can be updated or modified or scaled up as per the requirements of the lab sessions and can be chosen from the reference websites.)

Introduction: Open Source definition, open source technology importance in a perspective of Free and open Source Software (FOSS)

I: Introduction and syntax of Python programming
II: Python operators and Looping structures

III: Data Structures in Python

IV: Python Functions, Modules and Packages

V: Object Oriented Programming in Python

I. Experiment on basic control structures & loops.

a) Write a program for checking the given number is even or odd.

b) Using a for loop, write a program that prints the decimal equivalents of 1/2, 1/3, 1/4 ,....... 1/10

c) Write a program for displaying reversal of a number.

d) Write a program for finding biggest number among 3 numbers.

e) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

II. Experiment on operators & I/O operations.

a) Write a program that takes 2 numbers as command line arguments and prints its sum.

b) Implement python script to show the usage of various operators available in python language.

c) Implement python script to read person's age from keyboard and display whether he is eligible for voting or not.

d) Implement python script to check the given year is leap year or not.

III. Experiment on Python Script.

a) Implement Python Script to generate first N natural numbers.

b) Implement Python Script to check given number is palindrome or not.

c) Implement Python script to print factorial of a number.

d) Implement Python Script to print sum of N natural numbers.

e) Implement Python Script to check given number is Armstrong or not.

IV. Experiment on Lists.

a) Finding the sum and average of given numbers using lists.

b) To display elements of list in reverse order.

c) Finding the minimum and maximum elements in the lists.

d) Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file? e) Write a program to compute the number of characters, words and lines in a file.

V. Experiment on Strings.

a) Implement Python Script to perform various operations on string using string libraries.

b) Implement Python Script to check given string is palindrome or not.

c) Implement python script to accept line of text and find the number of characters, number of vowels and number of blank spaces in it.

VI. Experiment on functions.

a) Define a function max_of_three() that takes three numbers as arguments and returns the largest number

b) Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between 1000 and 2000.

c) Write a program to perform addition of two square matrices.

VII. Experiment on recursion & parameter passing techniques.

a) Define a function which generates Fibonacci series up to n numbers.

b) Define a function that checks whether the given number is Armstrong

c) Implement a python script for Call-by-value and Call-by-reference

d) Implement a python script for factorial of number by using recursion.

VIII. Experiment on Tuples.

a) Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number. Suppose the following input is supplied to the program: 34, 67, 55, 33, 12, 98. Then, the output should be: ['34', '67', '55', '33', '12', '98'] ('34',67', '55', '33', '12', '98').

b) With a given tuple (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), write a program to print the first half values in one line and the last half values in one line.

IX. Experiment on files.

a) Write Python script to display file contents.

b) Write Python script to copy file contents from one file to another.

X. Experiment on searching & sorting Techniques.

a) Implement a python script to check the element is in the list or not by using Linear search & Binary search.

b) Implement a python script to arrange the elements in sorted order using Bubble, Selection, Insertion and Merge sorting techniques.

XI. Experiment on Exception handling concepts.

a) Write a python program by using exception handling mechanism.

b) Write a python program to perform various database operations (create, insert, delete, update).

XII. Experiment on:

a) Write a program to calculate overtime pay of 10 employees. Overtime is paid at the rate of Rs.12.00 per hour for every hour worked above 40 hours. Assume that employee do not work for fractional part of an hour. Write a program to calculate overtime pay of 10 employees. Overtime is paid at the rate of Rs.12.00 per hour for every hour worked above 40 hours. Assume that employee do not work for fractional part of an hour.

b) Write a function that receives marks received by a student in 3 subjects and returns the average and percentage of these marks. Call this function from main() and print the result in main.

c) Write a program to demonstrate database connectivity in python.

d) Write a script that imports requests and fetch content from the page. Eg. (Wiki)

XIII. Experiments on python Framework.(Architecture of any python Framework (Flask, Django etc.)

a) Create a virtual environment and start a project by installing necessary packages

b) Connect Database with your project.

c) Generate HTML Forms with Form class and store data into the database.

d) Create a Word Counter in any Framework. (The counter will count the number of occurrence of each word in a paragraph).

e) Create an application to send an emails using any framework.

f) Create a Login System using any Framework.

g) Create an Online School System where teacher can create assignments that students can complete and view their results.

h) Create a Weather Application using any Framework and integrate it with some APIs (Application Program Interface).

i) introduction and small exercises on packages such as - Matplotlib (for the graph plotting), Tkinter (Python GUI programming package), Python web application using Flask, Web2py packages.

j) Introduction to Anaconda Navigator for python.

XIV. Mini Project on:

a) Develop front pages of a website showing introductory details of an organization.

XV. Mini Project on:

1. Develop a mini-project of students' choice to demonstrate creativity. Eg. music player, gamestation, student management systems, library management system etc.

a) Create a To-Do List app with registration, login, and CRUD Functionality.

b) Create a Chatting Application with Python.

c) Create a Token-based authentication system to work.

d) Create a Resume Builder and download that resume.

e) Create an app to take notes and store those notes in the backend database.

f) Automatic Tweet Posting

g) Railway Enquiry System.

h) Online Quiz Application

i) Ecommerce website, etc **Course Outcomes**

ITU429.1 Implement various applications using open source system of Python

ITU429.2 Create simple GUI applications and develop experiments using Python

ITU429.3 Understand configuration and virtual environment of open source systems and Python

ITU429.4 To be able to explain open source project structure and how to successfully setup a project