KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

B.Tech in COMPUTER SCIENCE AND ENGINEERING (AI & ML)

Course structure II YEAR (KR23)

Applicable from A.Y 2023-24 Batch

II YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	23CC401PC	Automata Theory and Compiler Design	3	0	0	3
2	23CC402PC	Data Structures	3	0	0	3
3	23CC403PC	Operating Systems	3	0	0	3
4	23CM404PC	Deep Learning	3	0	0	3
5	23CM405PC	Computer Organization and Architecture	3	0	0	3
6	23CC406PC	Data Structures Lab	0	0	2	1
7	23CC407PC	Operating Systems Lab	0	0	2	1
8	23CM408PC	Deep Learning Lab	0	0	2	1
9	23CM409PC	Real-time Research Project/Societal-related Research Project	0	0	4	2
10	*23MC410HS	Constitution of India	3	0	0	0
		Total	**18**	**0**	**10**	**20**

***MC – Satisfactory/Unsatisfactory**

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

B.Tech. in COMPUTER SCIENCE AND ENGINEERING (AI & ML)

II Year II Semester Syllabus (KR23)
AUTOMATA THEORY AND COMPILER DESIGN (23CC401PC)
Common to CSE, CSE (DS), IT, CSE (AI&ML)

Prerequisites / Corequisites:

L	T	P	C
3	0	0	3

1. 23CC304PC – Discrete Mathematics Course

Course Objectives: The course will help to

1. Provide introduction about central ideas of theoretical computer science from perspective of formal languages.
2. Introduce fundamental concepts of Formal languages, grammars, and types of automata.
3. Introduce the major concepts of language translation and compiler design.
4. Impart the knowledge of practical skills for constructing a compiler.
5. Introduce various phases of compiler with examples.

Course Outcomes: The student will be able to

1. Identify and understand the concept of abstract machines and their power to recognize the languages.
2. Explore context free grammar for formal languages.
3. Design a compiler on given a set of language features.
4. Implement LL and LR parsers.
5. Design and develop machine code from three address code.

UNIT-I:

Introduction to finite automata: The central concepts of automata theory, Structural representation of FA, Types of FA, Conversion of NFA to DFA and NFA with epsilon to NFA without epsilon.
Regular Expression: Introduction to Regular language and Regular Expression, Algebraic laws for regular expressions, Conversion of FA to RE and RE to FA, Pumping lemma for Regular languages.

UNIT-II:

Grammar: Definition of Grammar, Types of grammars, Derivation, derivation types, Derivation tree, Ambiguity, Left recursion and Elimination of Left recursion.
Push Down Automata (PDA): Definition, Structural representation, Construction of PDA, conversion of PDA – CFG and CFG to PDA.

UNIT-III:

Turing Machine (TM): Definition, Structural representation, Construction of Turing Machine on various languages.
Compiler: General Language processing system, Definition of Compiler, Phases of Compiler, Lexical Analysis, Input Buffering.

UNIT-IV:

Syntax Analysis: Introduction to types of Parsing techniques, Top-down Parsing: Recursive Decent parsing, Predictive parsing, Bottom-up Parsing: SLR, CLR, LALR.
Semantic Analysis: Introduction to Syntax Directed Definition, Syntax Directed Translation, Attributes, Types of Attributes, Bottom-up evaluation of attributes.
Intermediate code generation: Types of Intermediate codes, Types of Three address codes, Evaluation of three address code.

UNIT-V:

Runtime Environment: Storages organization, Storage allocation strategies: Static, Stack, Heap allocations, Activation Record.
Code Optimization: Introduction, principle sources of optimization, flowgraph, techniques in local loop and global optimizations.
Code generation: Issues in code generation, DAG, Simple code generator.

TEXT BOOKS:

1. Compilers: Principles, Techniques and Tools – Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, 2nd Edition, Pearson, 2007.
2. Introduction to Theory of Computation - Anil Maheshwari, Michiel Smid, April 17, 2019.
3. Compiler Construction – Principles and Practice, Kenneth.

REFERENCE BOOKS:

1. Modern compiler implementation in C – Andrew WAppel, Revised Edition, Cambridge University Press.
2. The Theory and Practice of Compiler writing- J.P. Tremblay and P.G. Sorenson, TMH.

B.Tech. in COMPUTER SCIENCE AND ENGINEERING (AI & ML)
II Year II Semester Course Syllabus (KR23)
DATA STRUCTURES (23CC402PC)
Common for CSE, CSE (DATA SCIENCE), CSE (AI&ML), IT

Course Objectives: The course will help to

L	T	P	C
3	0	0	3

1. Learn the syntax and semantics of the C++ programming language.
2. Implement ADTs such as lists, stacks, queues, trees, graphs, search trees in C++ to solve problems.
3. Choose an appropriate data structure for a specified application.
4. Understand the behaviour of data structures such as trees, hash tables, search trees, Graphs and their representations.
5. Understand and analyze various searching and sorting algorithms.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Understand C++ Program structure, functions, and templates.
2. Differentiate types of recursions, array and dynamic array and linear and non-linear data structures.
3. Construct programs using a variety of data structures such as stacks, queues, hash tables, binary trees, search trees, heaps, graphs.
4. Interpret appropriate data structures to represent data items in real world problems.
5. Design and implement sorting and searching algorithms, their implementation, efficiency, and practical application. They will be equipped to choose the most suitable algorithm for specific tasks and analyze the performance.

UNIT – I:

Basic Concepts of C++ - Structure of a C++ program, Data types, Declaration of variables, Expressions, Operators, Operator Precedence, Evaluation of expressions, Type conversions.

Flow control statement- if, switch, while, for, do, break, continue, goto statements. Functions - Scope of variables, Parameter passing, Default arguments.

Templates - Types of templates, Class - definition, structure, objects, access modifiers, scope, this pointer, Constructors and Destructors, inheritance, virtual functions.

UNIT – II:

Recursion, Arrays: Recursion, Direct Recursion, Indirect Recursion, Data Abstraction, Representation of single, two-dimensional arrays, row order majoring, column order majoring, Dynamic Array- polynomials, sparse matrices-array and linked representations, Dynamic Array vs Array.

Introduction to Linear data structures-Linear list ADT-array representation and linked representation, Types of Linked List - Singly Linked Lists-Operations-Insertion, Deletion, Doubly Linked Lists- Operations- Insertion, Deletion, Real Time Applications of Linked List.

UNIT- III:

Stacks: Definition, ADT, standard stack operations- array and linked list implementations, applications-infix to postfix conversion, postfix expression evaluation, parsing parentheses, reverse of a string using stack.

Queues: Definition, ADT, standard queue operations - array and linked implementations, Circular queues - Insertion and deletion operations.

UNIT IV:

Non-Linear Data Structures: Trees – Definition, terminology, Binary trees-definition, Properties of Binary Trees, Binary Tree ADT, representation of Binary Trees - array and linked representations, Binary Tree traversals- DFS-In-order, Post-order, Preorder, BFS – Level order traversal, Binary Search Tree ADT – BST traversal,

UNIT V:

Graphs–Definitions, Terminology, Applications and more definitions, Properties, Graph ADT, Graph Representations- Adjacency matrix, Adjacency lists, Graph Search methods - DFS and BFS.

Sorting- Merge Sort, Heap sort, **Priority Queues–**Definition and applications, Max Heap, Min Heap.

Hashing-Definition, hash tables, hash functions, Collision resolution techniques - linear probing, chaining.

TEXTBOOKS:

1. Data Structures Through C++ - Yashavant Kanetkar, 4th Edition, BPB Publications, 2022.
2. Data structures using C++- D. S. Malik, 2nd Edition, Cengage learning, 2009.
3. The Complete Reference C++- Herbert Schildt, 4th Edition, Tata Mc Graw Hill, 2017.

REFERENCE BOOKS:

1. Data Structures and Algorithm Analysis in C++, 4th Edition, Weiss Mark Allen, Pearson Education · 2014
2. The C++ Programming Language, 4th Edition, B. Stroutstrup, Pearson Education, 2013.

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

B.Tech. in COMPUTER SCIENCE AND ENGINEERING (AI & ML)

II Year II Semester Syllabus (KR23)

OPERATING SYSTEMS (23CC403PC)

Common to CSE, CSE (DS), CSE(AI & ML), IT

L	T	P	C
3	0	0	3

Prerequisites/ Corequisites:

1. 23CS103ES - Programming for Problem Solving Course.
2. 23CM405PC – Computer Organization and Architecture Course.

Course Objectives: The course will help to

1. Understand the design and the services provided by an operating system.
2. Learn different process scheduling algorithms and process synchronization.
3. Facilitate students in understanding Inter process communication along with deadlocks.
4. Categorize the operating systems' resource management techniques, file and memory managementtechniques.
5. Impart fundamentals of Disk Management and Protection.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Understand the basic concepts of Operating Systems and Linux Utility system calls.
2. Illustrate the different process scheduling algorithms and the challenges in process synchronization.
3. Demonstrate Inter process communication and deadlock.
4. Explore memory management techniques and file management concepts.
5. Analyze disk management and the importance of protection.

UNIT-I:

Introduction to Operating System – Operating system objectives, User view, System view, Operating system definition, OS Operations, Operating System services, OS Structures- Simple, Layered Architecture, Micro Kernel, Modular, Hybrid structure.

Introduction to Linux and Linux Commands - Architecture of LINUX, features of LINUX. Introduction to variousLINUX commands such as PATH, man, echo, printf, script, passwd, uname, who, date, pwd, cd, ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat.

UNIT-II:

Process and CPU Scheduling - Process concepts and scheduling, Operations on processes, Introduction to threads: Types and issues, Scheduling Criteria, Scheduling Algorithms: FCFS, SJF, SRTF/ SJN, Priority Non-Preemption, Priority Preemption, Round Robin, Multilevel feedback scheduling algorithms, etc.

System call interface for process management - Introduction to various system calls such as fork, vfork, exit, wait,waitpid, exec.

Process Management and Synchronization - The Race condition, The Critical Section Problem, Synchronization Hardware, Semaphores and Classical Problems of Synchronization, Critical Regions, Monitors.

UNIT-III:

Inter Process Communication Mechanisms - IPC using pipes, FIFOs, Semaphores: semget, semop, semctl, message queues: msgget, msgsnd, msgrcv, msgctl, shared memory: shmget, shmat, shmdt, shmctl, ipc status commands.

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

UNIT-IV:

Memory Management and Virtual Memory - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Demand Paging, Page Replacement, Page Replacement Algorithms: FIFO, LRU,Optimal.

File System Interface and Operations – Access methods: Direct Access, Sequential Access, Index Sequential Access, Directory Structure, Protection, File System Structure, **Allocation methods:** Contiguous Allocation, LinkedAllocation, Indexed Allocation, Free-space Management.

System calls for File Management - create, open, close, read, write, lseek, link, symlink, unlink, stat, fstat, lstat, chmod, chown, **Directory System calls** - opendir, readdir, closedir, mkdir, rmdir,umask.

UNIT-V:

Mass Storage Structure - Overview of Mass Storage Structure, Disk Structure, Disk Attachment.
Disk Scheduling: FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK, etc., Disk Management.
Protection – System Protection, Goals of Protection, Principles of Protection.
Domain of Protection: Access Matrix, Implementation of Access Matrix.

TEXT BOOKS:

1. Operating System Principles - Abraham Silberchatz, Peter B. Galvin, Greg Gagne, JohnWiley, 9thEdition,2013.
2. Advanced Programming in the UNIX Environment - W. Richard. Stevens, 3rd Edition, Pearson Education,New Delhi, India. 2013

REFERENCE BOOKS:

1. Internals and Design Principles - William Stallings, Operating Systems, 7 th Edition, Pearson education PHI, 2013.
2. Unix and shell Programming - Behrouz A. Forouzan, Richard F.Gilberg.Thomson, Cengage Learning.
3. UNIX Programming Environment - Kernighan and Pike, PHI/ Pearson Education.2015.

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

B.Tech. in COMPUTER SCIENCE AND ENGINEERING (AI & ML)

II Year II Semester Course Syllabus (KR23)

DEEP LEARNING (23CM404PC)

Pre-requisites/Co-requisites:

1. 23CM303PC – Artificial Intelligence Course.

L	T	P	C
3	0	0	3

Course Objectives: The course will help to

1. Introduce computer vision concepts and basics of Artificial Neural Networks.
2. Understand Image analysis and classification using CNN.
3. Explore advanced concepts of CNN and importance of transfer learning.
4. Expertise object detection using advanced techniques.
5. Generate new images using computer vision techniques.

Course Out comes: After learning the concepts of this course, the student is able to

1. Develop ANN application for tabular data.
2. Classify images using CNN.
3. Use advance CNN Architectures and transfer learning techniques.
4. Explore object detection techniques and generate images with masking of various classes.
5. Generate new images with GAN and other computer vision techniques.

Unit-I:

Computer Vision Fundamentals: Computer vision, Applications of computer vision, Computer vision pipeline: The big picture, Image input, Image preprocessing - Feature extraction - Classifier learning algorithm.

Deep learning and neural networks: Understanding perceptrons, Multilayer perceptrons, Activation functions, The feed forward process, Optimization algorithms, Back propagation.

Unit-II:

Convolution neural networks: Image classification using MLP, CNN architecture, Basic components of a CNN, Image classification using CNNs, Adding dropout layers to avoid over fitting, Convolution over color images (3D images), Case study: Image classification for color images.

Structuring DL projects and hyper parameter tuning: Defining performance metrics, Designing a baseline model Getting your data ready for training, Evaluating the model and interpreting its performance, Improving the network and tuning hyperparameters, Learning and optimization, Optimization algorithms, Regularization techniques to avoid overfitting, Batch normalization.

Unit-III:

Advanced CNN architectures: CNN design patterns, LeNet-5, AlexNet, VGGNet, Inception and GoogLeNet , ResNet

Transfer learning: Problems that transfer learning solve, Transfer learning, Transfer learning working, Transfer learning approaches, Choosing the appropriate level of transfer learning, Open source datasets, Case study: A pretrained network as a feature extractor.

Unit-IV:

Object detection with R-CNN, SSD, and YOLO: General object detection framework, Region-based convolutional neural networks(R-CNNs) , Single-shot detector (SSD) , You only look once (YOLO) .

Segmentation, Mask R-CNN and Instance Segmentation, U-Net and Semantic Segmentation Reducing Noise with **Auto encoders:** Creating a simple fully connected autoencoder, creating a convolutional autoencoder, Denoising images with autoencoders, Spotting outliers using Autoencoders, Variational Autoencoders, creating an inverse image search index with deep learning, Implementing a variational autoencoder.

Unit-V:

Generative adversarial networks: GAN architecture, Evaluating GAN models, Popular GAN applications.

DeepDream and neural style transfer, How convolutional neural networks see the world, DeepDream, Neural style transfer Deep Learning on Edge Devices with CPU/GPU, Optimization, Overview of deep learning on edge devices, Techniques used for GPU/CPU optimization, Overview of MobileNet .

TEXTBOOKS:

1. Deep Learning for Vision Systems by MOHAMED ELGENDY, Manning Publications Co, 2020.
2. Mastering Computer Vision with TensorFlow 2.x, Krishnendu Kar, Packt Publishing, 2020.
3. TensorFlow 2.0 Computer Vision Cookbook, Jesús Martínez, Packt Publishing, 2021.

REFERENCE BOOKS:

1. Advanced methods and Deep Learning in Computer Vision, edited by E.R. Davies, Mathew Turk, Academic Press, 2021.

B. Tech. in COMPUTER SCIENCE AND ENGINEERING(AI&ML)

II Year I Semester Syllabus (KR23)

COMPUTER ORGANIZATION AND ARCHITECTURE (23CM405PC)

L	T	P	C
3	0	0	3

Pre-requisites/ Co-requisites: NIL

Course Objectives: The course will help to

1. Understand the basic components of a computer, principles of computer organization and the architectural concepts.
2. Understand the instruction sets, instruction formats and various addressing modes.
3. Understand basic data representation and Computer arithmetic
4. Explore the memory Organization and I/O Organization.
5. Understand the concept of pipelining techniques.

Course Outcomes: The student will be able to

1. Understand the basics of instruction sets and their impact on processor design.
2. Demonstrate an understanding of the design of the functional units of a digital computer system.
3. Evaluate cost performance and design trade-offs in designing and constructing a computer processor including memory.
4. Design a pipeline for consistent execution of instructions with minimum hazards.
5. Recognize and manipulate representations of numbers stored in digital computers Use string processing in different applications

UNIT-I:

Digital Computers : Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture. Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.

Basic Computer Organization and Design : Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt.

UNIT- II:

Microprogrammed Control : Control memory, Address sequencing, micro program example, design of control unit.

Central Processing Unit : General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

UNIT- III:

Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation.

Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations. Decimal Arithmetic unit, Decimal Arithmetic operations.

UNIT- IV:

Input-Output Organization: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access.

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

UNIT- V:

Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics. Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor.

Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Inter processor arbitration, Inter-processor communication and synchronization, Cache Coherence.

TEXT BOOKS:

1. Computer System Architecture – M. Morris Mano, Third Edition, Pearson/PHI, 1992.

REFERENCE BOOKS:

1. Computer Organization – Carl Hamacher, Zvonks Vranesic, SafeaZaky, V th Edition, McGraw Hill, 2002.
2. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI, 2002.
3. Structured Computer Organization – Andrew S. Tanenbaum, 4 th Edition, PHI/Pearson, 2013.

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

B.Tech. in COMPUTER SCIENCE AND ENGINEERING (AI & ML)
II Year II Semester Course Syllabus (KR23)
DATA STRUCTURES LAB (23CC406PC)
Common for CSE, CSE (DATA SCIENCE), CSE (AI&ML), IT

L	T	P	C
0	0	2	1

Prerequisites/ Corequisites:

1. 23CS103ES - Programming for Problem Solving Course

Course Objectives: The course will help to

1. Introduce various concepts of C++ programming language.
2. Understand data structures such as stacks and queues.
3. Explore trees and graphs.
4. Introduce searching and sorting algorithms.
5. Implement Data Structures in real world problems.

Course Outcomes: After learning the concepts of this course, student is able to

1. Design and implement C++ programs for computing real life applications.
2. Understand basic elements of control statements, arrays, functions, pointers and strings, and data. structures like stacks, queues, and linked lists.
3. Implement searching and sorting algorithms.
4. Select appropriate Data Structure to solve the problems.
5. Test and debug the application.

List of Programs:

1. Write a program to create, insert an element, delete an element, traverse the linked list and find length of a singly linked list.
2. Write a program to create, insert an element, delete an element, traverse a doubly linked list and also traverse the doubly linked list in reverse order.
3. Write a program to create, insert an element, delete an element, traverse a circular linked list.
4. Write a program to implement stack operations using arrays and singly linked lists.
5. Write a program to convert an infix expression to postfix expression.
6. Write a program to evaluate a postfix expression.
7. Write a program to check whether an expression is having balanced parenthesis or not.
8. Write a program using stacks to check whether a String is a Palindrome or not.
9. Write a program to implement Queue operations using Arrays and singly linked lists.
10. Write a program to implement Circular Queue operations using Arrays and singly linked lists.

11. Write a program to implement a queue using two stacks.
12. Write a program to check if a queue can be sorted into another queue using a stack.
13. Write a program that implements the following sorting methods to sort a given list of integers in ascending order using merge sort and quick sort methods.
14. Write a program that implements Binary tree operations using arrays and doubly linked list.
15. Implement, for a binary tree, the inorder, preorder, postorder and level order traversal methods.
16. Write a program to find the second minimum node in a Binary tree.
17. Write a program to insert a node, delete a node and search for a node in a Binary Search Trees.
18. Write a program to create a graph using arrays and linked lists.
19. Implement the depth first search and breadth first search graph traversal methods.

TEXTBOOKS:

1. Data Structures Through C++ - Yashavant Kanetkar, 4th Edition, BPB Publications, 2022.
2. Data structures using C++- D. S. Malik, 2nd Edition, Cengage learning, 2009.
3. The Complete Reference C++- Herbert Schildt, 4th Edition, Tata Mc Graw Hill, 2017.

REFERENCE BOOKS:

1. Data Structures and Algorithm Analysis in C++, 4th Edition, Weiss Mark Allen, Pearson Education · 2014
2. The C++ Programming Language, 4th Edition, B.Stroutstrup, Pearson Education, 2013.

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

B.Tech. in COMPUTER SCIENCE AND ENGINEERING (AI & ML)

II Year II Semester Syllabus (KR23)

OPERATING SYSTEMS LAB (23CC407PC)

Common to CSE, CSE (DS), CSE (AI & ML), IT

L	T	P	C
0	0	2	1

Prerequisites/ Corequisites: NIL.

Course Objectives: The course will help to

1. Provide an understanding of the design aspects of operating system concepts through simulation.
2. Introduce basic Unix commands, system call interface for process management, Inter process Communication and I/O in Unix.
3. Demonstrate the knowledge of the components of computer and their respective roles in computing.
4. Recognize and resolve user problems with standard operating environments.
5. Gain practical knowledge of how programming languages, operating systems, and architectures interact and how to use each effectively.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Illustrate the use of Linux OS, by means of a command line shell.
2. Implement operating system concepts such as scheduling, deadlock management, file management and memory management.
3. Implement C programs using Unix system calls.
4. Demonstrate synchronization and various components of a typical operating system.
5. Analyze various system calls for managing processes, memory and the file system.

List of Exercises

Exercise 1:

Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral.

a. Disassemble and assemble the PC back to working condition.

b. Install MS windows on the personal computer.

c. Install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both windows and Linux.

Exercise 2:

a. Implement in c language the following Unix commands using system calls i) cat ii) ls iii) mv

b. Write a C program to create child process and allow parent process to display "parent" and the child to display "child" on the screen

Exercise 3:

Assume you have the following jobs to execute with one processor, with the jobs arriving in the order listed here:

i	T(pi)
0	80
1	20
2	10
3	20
4	50

with the following values write a program to get the required output which is listed below

a. Suppose a system uses FCFS scheduling. Create a Gantt chart illustrating the execution of these processes?
b. What is the average turnaround time for the processes?
c. What is the average wait time for the processes?

Exercise 4:

a. Write a C program that illustrate communication between two unrelated process using named pipes
b. Write a C program that receives a message from message queue and display them
c. Write a C program to allow cooperating process to lock a resource for exclusive use (using semaphore)
d. Write a C program that illustrate the suspending and resuming process using signal
e. Write a C program that implements producer-Consumer system with two process using semaphore

Exercise 5:

Consider the following snapshot of a system. P0, P1, P2, P3, P4 are the processes and A, B, C, D are the resource types. The values in the table indicates the number of instances of a specific resource (for example: 3 3 2 1 under the last column indicates that there are 3 A-type, 3 B-type, 2 C-type and 1 D-type resources are available after allocating the resources to all five processes). The numbers under allocation-column indicate that those number of resources are allocated to various processes mentioned in the first column. The numbers under Max- column indicate the maximum number of resources required by the processes. For example: in 1st row under allocation-column 2 0 0 1 indicate there are 2 A-type, 0 B-type, 0 C-type and 1 D- type resources are allocated to process P0. Whereas 4 2 1 2 under Max-column indicate that process P0's maximum requirement is 4 A- type, 2 B-type, 1 C-type and 2 D-type resources.

Process	Allocation A B C D	Max A B C D	Available A B C D
P0	2 0 0 1	4 2 1 2	3 3 2 1
P1	3 1 2 1	5 2 5 2	
P2	2 1 0 3	2 3 1 6	
P3	1 3 12	1 4 2 4	
P4	1 4 3 2	3 6 6 5	

Answer the following questions using banker's algorithm by providing all intermediate steps

a. How many instances of resources are present in the system under each type of a resource?
b. Compute the Need matrix for the given snapshot of a system.
c. Verify whether the snapshot of the present system is in a safe state by demonstrating an order in which the processes may complete. If a request from process P1 arrives for (1, 1, 0, 0), can the request be granted immediately?
d. If a request from process P4 arrives for (0, 0, 2, 0), can the request be granted immediately?

Exercise 6:

Write a C program to simulate the following memory management technique: Paging

Exercise 7:

a. Write a C program that takes one or more file/directory names as command line input and reports following information
File Type ii) Number of Links iii) Time of last Access iv) Read, write and execute permissions.
b. Write a C program to list every file in directory, its inode number and file name

TEXT BOOKS:

1. Operating System Principles - Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 9th Edition, JohnWiley, 2013.
2. Advanced Programming in the UNIX Environment -W. Richard. Stevens, 3rd Edition, Pearson Education, New Delhi, India.2013

REFERENCE BOOKS:

1. Internals and Design Principles - William Stallings, Operating Systems, 5th Edition, Pearson Education/PHI,2013.
2. Unix and shell Programming- Behrouz A. Forouzan, Richard F.Gilberg.Thomson.
3. UNIX Programming Environment, Kernighan and Pike, PHI/ Pearson Education, 2015.

Noted and approved.

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

B.Tech. in COMPUTER SCIENCE AND ENGINEERING (AI & ML)

II Year II Semester Course Syllabus (KR23)

DEEP LEARNING LAB (23CM408PC)

Pre-requisites/Co-requisites:

1. 23CM303PC – Artificial Intelligence Course.
2. 23CS207ES - Python Programming Lab Course

L	T	P	C
0	0	2	1

Course Objectives: The course will help to

1. Understand the fundamental concepts of deep learning, including neural networks, layers, activation functions, and backpropagation.
2. Design neural network architectures for various tasks such as image classification, object detection.
3. Understand the impact of hyperparameters on model performance and learn techniques for hyperparameter tuning.
4. Explore transfer learning techniques to leverage pre-trained models for specific tasks.
5. To Expertise on generating images with existing images, patterns and on generating images with masking.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Develop ANN applications for tabular data.
2. Implement neural network architectures for various tasks such as image classification, object detection.
3. Apply techniques for hyperparameter tuning to improve model performance.
4. Apply transfer learning techniques to leverage pre-trained models for specific tasks.
5. Generate images with masking of various classes.

LIST OF EXPERIMENTS:

1. Design single unit perceptron for classification of iris dataset without using predefined models.
2. Design, train and test the MLP for tabular data and verify various activation functions and optimizers tensorflow.
3. Design and implement to classify 32x32 images using MLP using tensorflow / keras and check the accuracy.
4. Design and implement a CNN model to classify multi category JPG images with tensorflow / keras and check accuracy. Predict labels for new images.
5. Design and implement a CNN model to classify multi category tiff images with tensorflow / keras and check the accuracy. Check whether your model is overfit / underfit / perfect fit and apply the techniques to avoid overfit and underfit like regulizers, dropouts etc.
6. Implement a CNN architecture (LeNet, Alexnet, VGG, etc) model to classify multi

category Satellite images with tensorflow / keras and check the accuracy. Check whether your model is overfit / underfit / perfect fit and apply the techniques to avoid overfit and underfit.

7. Implement ResNet model to classify multi category medical images with tensorflow / keras and check the accuracy. Check whether your model is overfit / underfit / perfect fit and apply the techniques to avoid overfit and underfit.
8. Implement an image classification model using transfer learning techniques and check accuracy. Tune the required hyperparameters.
9. Implement R-CNN model for object detection. Check with Fast and Faster R-CNN models.
10. Implement a model to mask various categories with Semantic Segmentation.
11. Implement an Autoencoder to de-noise image.
12. Implement a GAN application to convert images.

TEXTBOOKS:

1. Deep Learning for Vision Systems by MOHAMED ELGENDY, Manning Publications Co, 2020.
2. Mastering Computer Vision with TensorFlow 2.x , Krishnendu Kar, Packt Publishing, 2020.
3. TensorFlow 2.0 Computer Vision Cookbook, Jesús Martínez, Packt Publishing, 2021.

REFERENCE BOOKS:

1. Advanced methods and Deep Learning in Computer Vision, edited by E.R. Davies, Mathew Turk, Academic Press, 2021.