RAMAIAH
Institute of Technology

CURRICULUM

Academic Year 2025 - 2026

VII SEMESTER

B.E.

Computer Science and Engineering (AI&ML)

Batch -2022

RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
(Approved by AICTE, New Delhi & Govt. of Karnataka)
Accredited by NBA &NAAC with A Grade

MULTICORE ARCHITECTURE AND PROGRAMMING	
Course Code: CI71	**Credits: 3:0:1**
Prerequisite: Nil	**Contact Hours: 42L+14P**
Course Coordinator: Dr. Mohana Kumar S	

Course Contents

Unit I

Introduction to Multi-Core Architecture: Motivation for Concurrency in Software, Parallel Computing Platforms, Parallel Computing in Microprocessors, Differentiating Multi-Core Architectures from Hyper- Threading Technology, Multi-Threading on Single-Core Versus Multi-Core Platforms Understanding Performance, Amdahl's Law, Growing Returns: Gustafson's Law. System **Overview of Threading:** Defining Threads, System View of Threads, Threading Above the Operating System, Threads Inside The OS, Threads Inside the Hardware, What Happens When A Thread Is Created, Application Programming Models and Threading, Virtual Environment: VMs And Platforms, Runtime Virtualization, System Virtualization.

Unit II

Fundamental Concepts of Parallel Programming: Designing for Threads, Task Decomposition, Data Decomposition, Data Flow Decomposition, Implications of Different Decompositions, Challenges You'll Face, Parallel Programming Patterns, A Motivating Problem: Error Diffusion, Analysis of The Error Diffusion Algorithm, Alternate Approach: Parallel Error Diffusion, Other Alternatives.

Unit III

Threading and Parallel Programming Constructs: Synchronization, Critical Sections, Deadlock, Synchronization Primitives, Semaphores, Locks, Condition Variables, Messages, Flow Control-Based Concepts, Fence, Barrier, Implementation-Dependent Threading Features. Threading API's: Threading API's For Microsoft Windows, Win32/MFC Thread API's, Threading API's For Microsoft. NET Framework, Creating Threads, Managing Threads, Thread Pools, Thread Synchronization, POSIX Threads, Creating Threads, Managing Threads, Thread Synchronization, Signalling, Compilation and Linking.

Unit IV

Open MP: A Portable Solution For Threading: Challenges In Threading A Loop, Loop-Carried Dependence, Data-Race Conditions, Managing Shared And Private Data, Loop Scheduling And Portioning, Effective Use Of Reductions, Minimizing Threading Overhead, Work-Sharing Sections, Performance-Oriented Programming, Using Barrier And No Wait, Interleaving Single-Thread And Multi-Thread Execution, Data Copy-In And Copy-Out, Protecting Updates Of Shared Variables, Intel Task Queuing Extension to Open MP, OpenMP Library Functions, OpenMP Environment Variables, Compilation, Debugging, Performance.

Unit V

Solutions To Common Parallel Programming Problems: Too Many Threads, Data Races, Deadlocks, And Live Locks, Deadlock, Heavily Contended Locks, Priority Inversion, Solutions For Heavily Contended Locks, Non-Blocking Algorithms, ABA Problem, Cache Line Ping-Ponging, Memory Reclamation Problem, Recommendations, Thread-Safe Functions And Libraries, Memory Issues, Bandwidth, Working In The Cache, Memory Contention, Cache-Related Issues, False Sharing, Memory Consistency, Current IA-32 Architecture, Itanium Architecture, High-Level Languages, Avoiding Pipeline Stalls On IA-32,Data Organization For High Performance.

Text Books:

1. Multicore Programming, Increased Performance through Software Multi-threading by Shameem Akhter and Jason Roberts, Intel Press, 2006
2. Hennessey and Patterson: “Computer Architecture A Quantitative Approach”, 4th Edition, Elsevier, 2012.

Reference Book:

1. Kai Hwang, Naresh Jotwani: Advanced Computer Architecture - Parallelism, Scalability, Programmability, 2nd Edition, Tata McGraw Hill, 2011.

Course Outcomes (COs):

At the end of the course, the student will be able to:

1. Understand the concepts of parallel programming, Amdahl's law and threading. (PO1,2,3,5 PSO1)
2. Describe the challenges involved in designing multi threaded applications. (PO1,2,3,5 PSO1)
3. Apply OpenMP directives and techniques to develop parallel programs by managing threads. (PO1,2,3,5 PSO1)
4. Analyze synchronization , deadlock and memory related issues with respect to parallel processing. (PO1,2,3,5 PSO1)
5. Design and develop multithreaded applications to exploit multicore architecture. (PO1,2, 3,5,PSO1)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE): 50 Marks		
Assessment Tool	**Marks**	**Course outcomes addressed**
Internal test-I	30	CO1, CO2, CO3
Internal test-II	30	CO3, CO4, CO5
The average of the two internal tests shall be taken for 30 marks.		
Other components		
Lab component	20	CO1, CO2, CO3
Semester End Examination (SEE)	50	CO1, CO2, CO3, CO4, CO5

FOUNDATIONS OF COMPUTER VISION	
Course Code: CI72	**Credits: 2:1:0**
Prerequisite: Nil	**Contact Hours: 28L+14T**
Course Coordinator: Dr. Nithya N	

Course Contents

Unit I

Introduction: Computer vision, Imaging modalities, Fundamental steps in image processing, Applications of computer vision. Digital Image Fundamentals: Image formation model, Sampling and quantization, Relationships between pixels. Mathematical tools used in image processing.

Unit II

Spatial Filtering: Intensity transformation functions, Histogram processing (Histogram equalization, Histogram matching), Fundamentals of spatial filtering (Mechanisms of spatial filtering, correlation and convolution), Smoothing spatial filters, Sharpening spatial filters.

Unit III

Image Segmentation: Fundamentals, Detection of isolated points, line and basic edge, Thresholding, Region-based segmentation. Representation and Description: Representation (border following, chain codes, minimum-perimeter polygons) Boundary descriptors (simple descriptors, shape numbers), Region descriptors (simple descriptors, topological descriptors, texture).

Unit IV

Object Recognition: What Should Object Recognition Do? Feature, Geometric, and semantic questions, Patterns, and pattern classes, Recognition based on decision-theoretic methods, Matching, Optimum statistical classifier, Neural networks

Unit V

Morphological Processing: Erosion and Dilation, Opening and Closing, Hit-or-miss transform, Morphological algorithms (Boundary extraction, Hole filling, Extraction of connected components). Compression Techniques: Fundamentals, Compression methods (Huffman, Arithmetic, Run-length coding)

Text Book:

1. Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 3rd ed., Pearson, ISBN:9332570329, 2016.
2. Computer Vision: A modern approach, D.A. Forsyth, J.Ponce, Pearson Education, 2015

References:

1. Anil K. Jain, “Fundamentals of Digital Image Processing”, Pearson Education, 2001.

2. B. Chanda and D. Dutta Majumdar, “Digital Image Processing and Analysis”, PHI, 2003

Course Outcomes (COs):

At the end of the course, the student will be able to:

1. Understand the fundamentals of computer vision, various imaging modalities, and the mathematical tools used in image formation and analysis. (PO-1,2,5,8, PSO-1,2,3)
2. Apply spatial domain techniques such as intensity transformations, histogram processing, and filtering to enhance digital images. (PO-1,2,4,5,8, PSO-1,2,3)
3. Analyze image segmentation techniques and describe shapes/regions using appropriate boundary descriptors. (PO-1,2,3,4,5,8, PSO-1, 2,3)
4. Implement object recognition systems using feature extraction, pattern classification, and neural networks. (PO-1,2,3,4,5, PSO-1,2,3)
5. Perform morphological operations and evaluate image compression techniques for efficient image storage and processing. (PO-1,2,3,5,6,8, PSO-2,3)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE): 50 Marks		
Assessment Tool	**Marks**	**Course outcomes addressed**
Internal test-I	30	CO1, CO2, CO3
Internal test-II	30	CO3, CO4, CO5
The average of the two internal tests shall be taken for 30 marks.		
Other components		
Assignment 1	10	CO1, CO2, CO3
Assignment 2	10	CO1, CO2, CO3, CO4, CO5
Semester End Examination (SEE)	50	CO1, CO2, CO3, CO4, CO5

FULL STACK DEVELOPMENT	
Course Code: CIE733	**Credits: 3:0:0**
Prerequisite: Nil	**Contact Hours: 42L**
Course Coordinator: Subash N/Veena S	

Course Contents

UNIT I

Introduction to Full Stack Development: Client-server Architecture Static vs Dynamic websites Roles and responsibilities How front-end and back-end communicate Introduction, Cascading Styles Sheet: Concept of CS, Creating Style Sheet, CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts), CSS Id and Class, Box Model (Border, Padding, Margin properties), CSS Advanced (Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class)

UNIT II

Front-End Web Development: JavaScript syntax, Types of Data and Variables, Operations and calculations, The Document Object, Using Events. JavaScript Advanced: Scopes and Closures, understand "this" and prototypes, OOPs concepts as applied to JS and prototypal inheritance, Understanding the meaning of asynchronous. Event loops, Promises.

UNIT III

ReactJS: Building blocks of React, Create-react-app - Create first React app using this CLI, JSX - Understand what it is and how it’s required to create components, Simple functional components, CSS - Load CSS and use it via class Name, props - Passing props to components to make them reusable

Unit IV

Back-End Web Development: Node.js: Introduction - What is Node.js, Architecture, Feature of Node JS, Installation and setup - Creating webservers with HTTP (Request & Response), Understand dependence management: npm and package.json File system APIs. CRUD Operations using Node.js: Event Handling - GET & POST implementation.

Unit V

Database Integration: Overview of NoSQL vs SQL Document-oriented database model Key features of MongoDB, Core MongoDB Operations (CRUD) Schema Design & Data Modeling NoSQL MongoDB Database using Node.js, Implementation of CRUD operations

Text Books:

1. Steven A. Gabarro, “Web Application Design and Implementation: Apache 2,PHP5, MySQL, JavaScript, and Linux/UNIX”, Wiley- IEEE Computer SocietyPress 2007.

2. Nate Murray, Felipe Coury, Ari Lerner and Carlos Taborda, “Ng-book, The Complete Book on Angular”, Fullstack.IO, 1st edition, 2016.

3. KrasimirTsonev, “Node.js by Example”, Packt Publishing Limited, 2015.

References:

1. Web link for Node.js: https://nodejs.org/en/
2. Web link for MongoDB: https://www.mongodb.com/
3. Web link for React JS: https://reactjs.org/

Course Outcomes (COs):

At the end of the course, the student will be able to:

1. Understand the client-server architecture and design responsive web pages using CSS styling techniques. (PO-1,2,3,5, PSO-1,2)
2. Develop interactive front-end applications using JavaScript and object-oriented features. (PO-1,2,3,5, PSO-1,2)
3. Implement reusable user interfaces using ReactJS by leveraging components, JSX, props, and event-driven programming. (PO-1,2,3,5, PSO-1,2)
4. Demonstrate server-side functionality using Node.js by creating web servers and HTTP requests. (PO-1,2,3,5, PSO-1,2)
5. Integrate NoSQL databases and perform CRUD operations for full-stack applications. (PO-1,2,3,5, PSO-1,2)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE): 50 Marks		
Assessment Tools	**Marks**	**Course Outcomes (COs) addressed**
Internal Test-I (CIE-I)	30	CO1, CO2
Internal Test-II CIE-II)	30	CO3, CO4, CO5
Average of the two CIE shall be taken for 30 marks		
Other Components		
Programming Assignment-1	10	CO1, CO2, CO3
Programming Assignment-2	10	CO4, CO5
The Final CIE out of 50 Marks = Average of two CIE tests for 30 Marks+ Marks scored in Case Study +Marks scored in Assignment		
Semester End Examination (SEE)	50	CO1, CO2, CO3 CO4, CO5

CONTAINERIZATION LABORATORY	
Course Code:CIL74	**Credits: 0:0:1**
Prerequisite: Nil	**Contact Hours: 14P**
Course Coordinator: Dr. Siddesh GM	

Course Contents

1. Overview of virtualization, AWS Platform, Load Balancing
2. Amazon Virtual Private Cloud.
3. Introduction to GIT Workflow.
4. Working with various commands in GIT.
5. Recording Changes to the Repository.
6. Creating a Service in Kubernetes, Installing Kubernetes Dashboard.
7. Deploying an App using Dashboard.
8. Using Rolling Updates in Kubernetes.
9.Containers and Container Orchestration.
10. Working with Docker Containers.
11.Docker Command Line Interphase.
12.Working with images using Dockers

References:

1. Chris Richardson: Microservices Patterns with Examples in Java, Manning Publications Co., First Edition, 2019.
2. Moises Macero: Learn Microservices with Spring Boot: A Practical Approach to RESTful Services using RabbitMQ, Eureka, Ribbon, Zuul and Cucumber, A Press, First Edition, 2017.
3. Sourabh Sharma: Mastering Microservices with Java 9, Packt Publishing Ltd, Second Edition, 2017.
4. Amazon Web Services in Action – Michael Wittig & Andreas Wittig (Manning, 3rd Ed.)2. AWS Certified Solutions Architect Official Study Guide – Ben Piper & David Clinton (Sybex).
5. Pro Git – Scott Chacon & Ben Straub (Apress, 2nd Ed., free online at git-scm.com/book).
6. Kubernetes Up & Running – Kelsey Hightower, Brendan Burns, Joe Beda (O'Reilly, 3rd Ed.).
7. Kubernetes Patterns – Bilgin Ibryam & Roland Huß (O'Reilly).
8. Docker Deep Dive – Nigel Poulton (Leanpub, latest edition).

Course Outcomes (COs):

At the end of the course, the student should be able to:

1. Explain the concepts of virtualization and the fundamentals of load balancing in cloud environments. (PO-1,2,3,4,5,PSO-1,2)
2. Demonstrate proficiency in version control using Git by applying core commands and workflows for tracking and managing source code. (PO-1,2,3,4,5,PSO-1,2)
3. Deploy and manage containerized applications using Docker, Kubernetes including image handling and command-line operations. (PO-1,2,3,4,5,PSO-1,2)

Course Assessment and Evaluation:

Continuous Internal Evaluation (CIE): 50 Marks		
Assessment Tools	**Marks**	**Course Outcomes (COs) addressed**
Lab Test-I	10	CO1, CO2
Lab Test-II	10	CO3
Weekly Evaluation + Lab Record	30	CO1, CO2, CO3
The Final CIE out of 50 Marks = Marks of Lab Record + Marks scored in Lab Test-I + Marks scored in Lab Test-II		
Semester End Examination (SEE)	50	CO1, CO2, CO3

SKILL ENHANCEMENT LABORATORY- GENERATIVE AI	
Course Code:CI75	**Credits: 0:1:1**
Prerequisite: Nil	**Contact Hours: 14T+14P**
Course Coordinator:	

Course Contents

- Introduction to Generative AI
- Image and Video Generation Tools
- Copilot, Claude, BRAD, Educational generative AI.
- Preliminary Project Based Learning
- Large Language Models Prompt Engineering
- DNNs Tensor Flow
- Word Embedding’s
- Hugging Face Models
- LangChain - Basics of Chains
- LangChain Model IO, LangChain - RAG
- LangChain – Memory, LangChain - Agents
- Recurrent neural network, Long Short-Term Memory
- Transformers
- Project Based Learning

The list of Laboratory Exercises formulated covering all the above topics are mentioned in the table below:

Expt. No.	Laboratory Exercises
1.	Using Generative AI tools, create a short story or poem (text generation) based on a chosen theme, then generate an image (text-to-image generation) and a 15–30 second sound or music clip (AI sound generation) that represent the same theme, and analyzing how well the text, image, and sound outputs align with each other while discussing the strengths, limitations, and ethical challenges of multimodal generative AI.
2.	Use word embeddings to improve prompts for Generative AI model. Retrieve similar words using word embeddings. Use the similar words to enrich a GenAI prompt. Use the AI model to generate responses for the original and enriched prompts. Compare the outputs in terms of detail and relevance.
3.	Develop a predictive model for academic performance. Use TensorFlow to build a DNN that predicts whether a student will pass or fail based on their study hours and attendance percentage.
4.	Your department wants a tool to summarize long academic reports into concise summaries for easier review. Use a pre-trained Hugging Face summarization model to generate a short summary from a given academic report.
5.	i. Explore pretrained word vectors. Explore word relationships using vector arithmetic. Perform arithmetic operations and analyze results ii. Use dimensionality reduction (e.g., PCA or t-SNE) to visualize word embeddings for the above pretrained word vectors. Select 10 words from a specific domain (e.g., sports, technology) and visualize their embeddings. Analyze clusters and relationships. Generate contextually rich outputs using embeddings. Write a program to generate 5 semantically similar words for a given input.
6.	Train a custom Word2Vec model on a small dataset. Train embeddings on a domain-specific corpus (e.g., legal, medical) and analyze how embeddings capture domain-specific semantics.
7.	Install langchain , cohere (for key) and langchain-community. Using the API key, load a text document from the your google drive . Create a prompt template to display the output in a particular manner.
8.	Use a pre-trained Hugging Face model to analyze sentiment in text. i. Assume a real-world application ii. Load the sentiment analysis pipeline. iii. Analyze the sentiment by giving sentences to input.
9.	Create a LangChain application that takes the name of an institution as input and extracts structured information from Wikipedia. Use Pydantic to define the schema for the desired output and implement a custom output parser to enforce the schema. Your task is to invoke the chain and fetch results about the institution with the following details: i. The founder of the institution

	ii. The year it was founded iii. The current branches of the institution iv. The number of employees working in it v. A concise 4-line summary of the institution
10.	Download the official Indian Penal Code (IPC) document in text or PDF format. Using LangChain (or a similar framework) and a language model of your choice (e.g., Cohere, OpenAI, or Hugging Face models), build a chatbot that can interact with the IPC. The chatbot should allow users to ask questions about specific sections, laws, or provisions, and provide relevant answers. Implement the following steps: i. Load and preprocess the IPC document (text extraction if PDF). ii. Split the document into smaller chunks for efficient retrieval. iii. Create an embedding-based vector store to enable semantic search. iv. Integrate a retrieval-based QA pipeline with a language model. v. Deploy a chatbot interface where users can query the IPC and receive conversational.
11.	Write a PyTorch program that performs time series forecasting using an RNN. The program should complete the following steps: i. Generate synthetic time series data using a sine function with added noise and visualize the data using matplotlib. ii. Prepare the dataset by creating input–output pairs using a fixed time step window (sequence length). iii. Define an RNN model using PyTorch, including: a recurrent layer (nn.RNN) a fully connected layer (nn.Linear) for prediction. iv. Train the model using the Mean Squared Error (MSE) loss function and the Adam optimizer. Use the trained model to make predictions on the time series data. v. Print the loss value during training to monitor progress. vi. Visualize the original time series data along with the predicted values to assess the model's performance.
12.	Build and train an LSTM-based text generation model using a dataset of your choice (e.g., Shakespeare's plays, song lyrics, or news articles). Preprocess the text into sequences, train the LSTM to predict the next character or word, and then generate new text by providing a starting seed phrase. Experiment with hyper-parameters such as sequence length, number of LSTM units, and training epochs.

Text Book:

1. Venkata Reddy Konasani, Shailendra Kadre. Machine Learning and Deep Learning Using Python and TensorFlow 1st Edition, McGraw Hill, 2021
2. Raghav Bali. Generative AI with Python and TensorFlow 2. Packt Publishing, 2021.

Course Outcomes:

1. Apply fundamental concepts of Large Language Models (LLMs), Prompt Engineering, and Word Embeddings to optimize AI-driven text processing tasks. (PO – 1, 2, 3,4, 5, 9) (PSO – 2)
2. Implement and evaluate Artificial Neural Networks (ANNs) and Deep Neural Networks (DNNs) using TensorFlow for efficient model training and performance tuning. (PO – 1, 2, 3,4, 5, 9) (PSO – 2)
3. Develop AI applications using Hugging Face Models, LangChain (Chains, Model IO, RAG, Memory, Agents), RNNs, LSTMs, and Transformers to enhance intelligent system capabilities. (PO – 1, 2, 4, 5) (PSO – 2)

Continuous Internal Evaluation

Continuous Internal Evaluation (CIE): 50 Marks		
Assessment tool	**Marks**	**Course outcomes attained**
Weekly Assessment	30	CO1, CO2, CO3
Project Evaluation (Internal)	20	CO1, CO2, CO3
Semester End Evaluation		
Project Evaluation (External)	50	CO1, CO2. CO3

Sl. No	**Evaluation Criteria for Project**	**Marks**
1	Problem Definition & Motivation	5
2	Literature Review & Technology Selection	5
3	Design, Methodology & Implementation	15
4	Results, Evaluation & Analysis	10
5	Demo, Presentation & Documentation	10
6	Innovation & Creativity	5