:

KRISHNA UNIVERSITY

BOARD OF STUDIES (BoS) – DEPARTMENT OF COMPUTER SCIENCE

MINUTES OF THE BOARD OF STUDIES MEETING

Agenda:

Framing and Approval of Curriculum and Syllabus for I, II, III and IV Semesters of Diploma Programmes

- Diploma in Computer Applications
- Diploma in Artificial Intelligence and Robotics
- Diploma in Artificial Intelligence and Machine Learning
- Diploma in Cybercrime and Cyber Security
- Diploma in Data Analytics

Members Present:

S. No.	Name	Designation	Role
1	Prof. R. Kiran Kumar	Dept. of Computer Science, Krishna University	Chairman
2	Smt. P. Durga Bhavani	HOD, Dept. of Computer Science, Nalanda Degree College, Vijayawada	Member
3	Kondapalli Ramkumar	HOD, Dept. of Electronics and Internet of Things, Nalanda Degree College, Vijayawada	Member
4	Bommareddy Sudheer Reddy	Lecturer, Dept. of Computer Science, Nalanda Degree College & PG Centre, Vijayawada	Member
5	Prof. P. Suresh Varma	Dept. of CSE, Adikavi Nannaya University, Rajahmundry	Member
6	Dr. M.H.M. Krishna Prasad	Dept. of CSE, JNTUK, Kakinada	Member
7	Prof. D. Lalitha Bhaskari	Dept. of CS & Systems Engineering, Andhra University	Member
8	Assoc. Prof. M. Babu Reddy	HOD, Dept. of Computer Science, Krishna University	Member

Proceedings

The meeting commenced with the Chairman welcoming all the members. The proposal for introducing the above **Diploma programmes** was placed before the Board for detailed discussion.

The members discussed the proposed **curriculum, syllabus course structure, credits, teaching scheme, practical components, Course Outcomes, and assessment pattern** of the five Diploma programmes.

The Board members emphasized the importance of providing students with **practical, skill-oriented, and industry-relevant knowledge** in emerging areas of Computer Applications, Artificial Intelligence, Robotics, Machine Learning, Cyber Security, Cybercrime, and Data Analytics.

The members also suggested incorporating adequate **laboratory work, hands-on training, project-based learning, and application-oriented components** to enhance the employability and technical skills of students.

Resolution

After detailed discussion, the **Board of Studies approved/recommended the introduction of the following Diploma programmes**:

1. **Diploma in Computer Applications**
2. **Diploma in Artificial Intelligence and Robotics**
3. **Diploma in Artificial Intelligence and Machine Learning**
4. **Diploma in Cybercrime and Cyber Security**
5. **Diploma in Data Analytics**

The Board recommended that the proposed **curriculum and syllabus , course structure, credits, Course Outcomes, teaching scheme, practical components, and assessment pattern** be finalized by incorporating the suggestions given by the members.

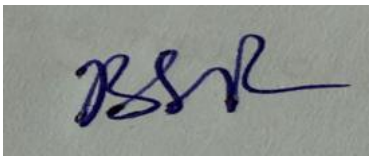
The Board further recommended that the finalized curriculum and syllabi be submitted to the **competent academic authority of Krishna University** for approval and implementation.

Conclusion

The Chairman thanked all the members for their valuable suggestions and active participation in the discussion. The meeting concluded with a vote of thanks.

Chairman, Board of Studies
Prof. R. Kiran Kumar
Signature: _____











Model Syllabus for Diploma Course in Artificial Intelligence and Robotics

COURSE STRUCTURE (for Semester I to IV)

Year	Semester	Course	Title of the Course	No. of Hrs / Week	No. of Credits
I	I	1	Python Programming and Data Structures	3	3
			Python Programming and Data Structures practical	2	1
	II	2	Artificial & Computational Intelligence	3	1
			Artificial & Computational Intelligence Practical	2	1
II	III	3	CONVERSATIONAL AI	3	1
			CONVERSATIONAL AI practical	2	1
	IV	4	Robotics Principles & Embedded systems	3	1
			Robotics Principles & Embedded systems Practical	2	1

SEMESTER-I

COURSE 1 : PYTHON PROGRAMMING AND DATA STRUCTURES

Theory **Credits: 3** **3 hrs/week**

Course Objectives

1. To introduce the fundamentals of Python programming, including environment setup, syntax, and core concepts.
2. To develop problem-solving skills using control flow, functions, and modules.
3. To provide knowledge of Python data structures, file handling, and exception handling for effective programming.
4. To impart object-oriented programming concepts and GUI development skills for building applications.

Course Outcomes (COs)

After successful completion of the course, students will be able to:

1. Explain the basic features, syntax, data types, and operators of Python programming.
2. Apply control flow constructs, functions, and modules to develop structured Python programs.
3. Demonstrate the use of sequences, sets, and dictionaries for effective data handling and manipulation.
4. Implement file handling techniques and apply exception handling mechanisms for robust applications.
5. Develop object-oriented and GUI-based applications using Python.

Unit 1: Basics of Python Programming:

Introduction to Python, Features of Python, Programming Modes - Interactive Mode & Script Mode, Identifiers, Naming Conventions, Keywords (Reserved Words), Built-in Data Types, Literals - Integer, Float, Complex, Boolean, String, Variables, Operators, Expressions, Assignment Statements, Input/Output Statements, Python Syntax (Lines, Comments, Indentation),

Operators & Operands, Classification of Operators - Arithmetic Operators, Relational Operators, Logical Operators, Bitwise Operators, Assignment, Augmented Assignment, Identity Operators, Expressions & Precedence Rules

Unit 2: Control Flow, Functions & Modules:

Control Flow - if Statement, if-else, if-elif-else. Iterative Statements – while, for, Nested Loops, Loop Control Statements – break, continue, pass; else with loops

Need for Functions, Defining & Invoking User-defined Functions, Return Statement, Function Input/Output Cases, Scope of Variables - Local, Global, Nested Functions, Function Arguments - Required, Positional, Default, Variable-length, main() Function, Documentation Strings, Recursive Functions, Anonymous Functions (Lambda), Library Functions

Modules - Import, from..import, Creating & Using Modules, Namespaces

Unit 3: Core Data Types and Python Collections

Strings: Representation, Indexing, Slicing, Immutability, Operators, Methods, Formatting

Lists: Creation, Indexing, Slicing, Mutability, Common Methods, List Comprehension

Tuples: Immutability, Operations, Tuple Assignment

Sets and Frozensets: Methods, Mathematical Operations, Comprehension

Dictionaries: Key-Value Structure, Methods, Traversal, Nested Dictionaries

Unit 4: File Handling, Exception Management & Object-Oriented Programming

File Handling: Types, Opening, Reading, Writing, Closing, CSV Files, OS/Pathlib

Error Types, Exception Handling: try-except, raise, User-defined Exceptions, Assertions

OOP Concepts: Classes, Objects, Attributes, Methods, Constructor and Destructors

Encapsulation: Private and Public Members

Inheritance: Single, Multilevel, Multiple, Method Overriding

Unit 5: Abstract Data Structures and GUI Programming

Abstract Data Structures (ADTs): Concepts and Importance

Linked Lists: Definition, Types- Singly, Doubly, Circular; Node Structure, Insertion, Deletion, Traversal (Single Linked list implementation only)

Stacks: LIFO Principle, Implementation using List, Applications

Queues: FIFO Principle, Implementation using List, Priority Queues

GUI Programming with Tkinter: Widgets (Label, Button, Entry, Menu, Listbox, Canvas etc.), Event Handling, Building Simple GUI Apps

Textbooks:

1. Python Programming-An Object Oriented approach, Anita Goel, Universities Press
2. Python Programming using Problem Solving Approach Reema Thareja Oxford University Press 2020
3. Exploring Python, Budd T A, McGraw-Hill Education, 1st Edition, 2011.

Reference Book:

1. Python: The Complete Reference, Martin C. Brown, Mc Graw-Hill, 2018
2. Fundamentals of Python, Kenneth A. Lambert. (2019), First Programs,2nd Edition, CENGAGE Publication.

Activities:

Outcome: Explain the basic features, syntax, data types, and operators of Python programming.

Activity: Conduct a "Python Basics Lab" where students write small programs to demonstrate literals, variables, data types, and operators (e.g., swapping numbers, simple calculator).

Evaluation Method:

- Lab performance checklist (execution of 3 mini tasks)
- Short quiz with multiple-choice and fill-in-the-blanks on syntax, data types, and operators

Outcome: Apply control flow constructs, functions, and modules to develop structured Python programs.

Activity: Group activity - "Python Problem Solving Challenge": Students solve real-life problems (e.g., finding prime numbers, grade calculator, menu-driven calculator) using control structures, functions, and importing standard modules.

Evaluation Method:

- Code submission with proper use of functions/modules (20%)
- Viva-voce to explain logic and flow of control (40%)
- Unit test with scenario-based programming questions (40%)

Outcome: Demonstrate the use of sequences, sets, and dictionaries for effective data handling and manipulation.

Activity: Hands-on mini project – "Student Data Manager": Students create a program using lists, tuples, sets, and dictionaries to store and manipulate student records (e.g., marks, courses, hobbies).

Evaluation Method:

- Practical demo of program with at least 5 data operations (add, search, delete, update, traverse)
- Evaluation rubric for correctness, efficiency, and use of appropriate data structure

Outcome: Implement file handling techniques and apply exception handling mechanisms for robust applications.

Activity: Individual assignment - "File-Based Address Book": Students create a program to store, update, and retrieve data from files, with exception handling for invalid inputs or missing files.

Evaluation Method:

- Assessment of program correctness (file read/write, append, delete, exception handling)
- Short quiz with error-tracing and debugging questions (given code with errors, students identify and correct)

Outcome: Develop object-oriented and GUI-based applications using Python.

Activity: Mini Project – "Student Information System with GUI": Students design a simple Tkinter-based application with classes/objects for handling student data, including basic GUI widgets (Entry, Button, Listbox).

Evaluation Method:

- Project demo and presentation (50%)
- Rubric-based evaluation for OOP concepts (classes, inheritance, encapsulation) and GUI design (widgets, event handling) (30%)
- Peer review/feedback on usability (20%)

SEMESTER-I

COURSE 1 : PYTHON PROGRAMMING AND DATA STRUCTURES

Practical **Credits: 1** **2 hrs/week**

1. Basic Python Programs:
    a. Write a program to display basic details (name, roll number, department) using print() and demonstrate different literal types (int, float, string, boolean, complex).
    b. Write a program to perform arithmetic, relational, logical, bitwise, and assignment operations on given inputs.
2. Control Flow Practice
    a. Write a program to find the largest of three numbers using if-elif-else.
    b. Write a program to check whether a number is prime or not using loops.
    c. Write a program to illustrate the use of loop control statements (break, continue, pass).
3. Functions and Recursion
    a. Write a program to define a function to calculate factorial of a number (using recursion).
    b. Write a program to demonstrate different types of function arguments (default, positional, keyword, variable-length).
4. Write a program to illustrate string slicing, concatenation, repetition, and built-in methods.
5. Write a program to create a list of numbers, perform insertion, deletion, searching, sorting, and list comprehension.
6. Write a program to demonstrate tuple packing, unpacking, and immutability.
7. Write a program to implement set operations (union, intersection, difference, subset, superset).
8. Write a program to create a dictionary of student roll numbers and marks, and perform add, update, delete, and traversal operations.
9. Write a program to read and display count of vowels, consonants, digits, and spaces of a text file.
10. Write a program to copy the contents of one file into another file.
11. Write a program to read and process student marks from a CSV file (calculate average, highest, lowest).

12. Write a program to demonstrate exception handling using try-except-finally.
13. Write a program to create a class Student with attributes and methods to display details.
14. Write a program to demonstrate single and multilevel inheritance.
15. Implement stack (LIFO) and queue (FIFO) using lists.
16. Implement singly linked lists: node creation, insertion, deletion, traversal.
17. Write a Tkinter program with Label, Entry, and Button widgets to take user input and display it.
18. Write a Tkinter program to create a simple calculator application.

SEMESTER-II

COURSE 2 : ARTIFICIAL & COMPUTATIONAL INTELLIGENCE

Theory **Credits: 3** **3 hrs/week**

Course Objectives

- Introduce foundational concepts and history of Artificial Intelligence (AI).
- Teach the PEAS framework and agent-based problem solving.
- Develop understanding of uninformed and informed search methods.
- Provide basic knowledge of machine learning concepts and computational intelligence.
- Instill awareness of ethical considerations in AI.
- Introduce logic programming fundamentals using Prolog as a practical tool.

Course Outcomes

Students completing this course will be able to:

- Explain basic AI terminology, agent models, and the PEAS framework.
- Formulate AI problems using search strategies and implement basic algorithms conceptually.
- Understand core machine learning categories and their applications.
- Describe computational intelligence approaches and their role in AI.
- Analyze ethical issues related to AI technologies.
- Write and execute simple Prolog programs demonstrating logic programming fundamentals.

Unit 1: Introduction to Artificial Intelligence and PEAS Framework

Introduction to AI: Definition, history, applications, and scope

The PEAS framework: Performance Measure, Environment, Actuators, Sensors, Examples of PEAS in real-world AI systems

Intelligent agents: Intelligent agents and their environments, **Types of intelligent agents:** Simple reflex, model-based, goal-based, utility-based, Agent architectures and rationality

Unit 2: Expert Systems

Definition and components of Expert Systems (Knowledge Base, Inference Engine, User Interface), Rule-based systems and knowledge representation, Examples of expert systems: medical diagnosis, decision support, Limitations and comparison with AI agents, Role and significance of expert systems in AI evolution

Unit 3: Search Strategies in AI

Problem-solving as search: problem formulation, states, actions, goal test, Traveller's problem

Uninformed (Blind) Search: Breadth-first search, Depth-first search, Uniform-cost search

Informed (Heuristic) Search: Greedy best-first search, A* algorithm

Applications of search in AI problems

Unit 4: Introduction to Machine Learning

What is machine learning, definitions, Types of learning: Supervised, Unsupervised, Reinforcement learning (basic ideas), classification, Regression , clustering and Association, Basic learning algorithms overview and applications

Unit 5: Computational Intelligence and Ethics in AI

Overview of computational intelligence (Basics of fuzzy logic, neural networks), Role of computational intelligence in AI, Ethics and societal challenges in AI, Responsible AI, fairness, transparency, and safety concerns

Unit 1: Introduction to AI, PEAS Framework & Intelligent Agents

Activities:

Interactive lecture with real-world AI examples

Group discussion on PEAS framework for different environments

Case study analysis of intelligent agents in everyday AI applications

Quiz on AI fundamentals and agent types

Outcomes:

Students will explain AI basics and describe the PEAS components.

Students will identify types of intelligent agents and their roles.

Evaluation Method:

Quiz and participation: 10%

Assignment on PEAS and agent modeling: 10%

Unit 2: Expert Systems

Activities:

Lecture with multimedia explaining components of expert systems

Hands-on group activity designing rule-based systems for simple decision problems

Case study review of medical diagnosis expert systems

Class debate on limitations and advantages of expert systems

Outcomes:

Understand expert system architecture and knowledge representation.

Develop simple rule-based systems for decision making.

Analyze real-world expert system examples critically.

Evaluation Method:

Group assignment designing rule base: 15%

Written test on expert system concepts: 10%

Unit 3: Search Strategies in AI

Activities:

Demonstration of uninformed search algorithms

Interactive exercises formulating search problems

Simulation of heuristic search and A* algorithm

Problem-solving sessions applying search to puzzles

Outcomes:

Formulate search problems and apply uninformed and heuristic algorithms conceptually.

Explain the working and use cases of different search strategies.

Evaluation Method:

Problem formulation assignment: 10%

Class test on search algorithms: 10%

Unit 4: Basics of Machine Learning

Activities:

Video lectures on different machine learning paradigms

Simple data classification exercises/classification demos

Group presentations on various ML types and applications

Quiz on ML types and terminology

Outcomes:

Explain supervised, unsupervised, and reinforcement learning basics.

Identify sample use cases for machine learning.

Evaluation Method:

Quiz and presentation: 10%

Written assignment on ML overview: 5%

Unit 5: Computational Intelligence & Ethics

Activities:

Lecture introducing fuzzy logic, neural networks overview

Ethical dilemma discussions and case studies in AI technology

Role play on AI responsibility and fairness

Group project: Develop guidelines for responsible AI use

Outcomes:

Understand computational intelligence basics and ethical AI challenges.

Propose responsible AI practices for diverse scenarios**.**

Evaluation Method:

Participation in ethics discussions: 5%

Group project report and presentation: 15%

Recommended Textbooks and References

- Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition
- Elaine Rich, Kevin Knight, Artificial Intelligence, 3rd Edition
- Michael Negnevitsky, Artificial Intelligence: A Guide to Intelligent Systems
- Ivan Bratko, Prolog Programming for Artificial Intelligence, 4th Edition
- Online resources: AI course materials from Coursera, NPTEL, GeeksforGeeks AI tutorials.

SEMESTER-II

COURSE 2 : ARTIFICIAL & COMPUTATIONAL INTELLIGENCE

Practical **Credits: 1** **2 hrs/week**

***Note**: Experiments have to be conducted using Prolog*

1. Introduction to Prolog Environment and Syntax
   - Setting up Prolog, understanding facts, rules, and queries.
2. Defining Simple Facts and Queries
   - Write and test simple facts like family relationships, likes/dislikes.
3. Creating Rules in Prolog
   - Define logical rules with conditions and test queries.
4. List Handling in Prolog
   - Write programs to manipulate lists (head, tail, concatenation).
5. Recursion in Prolog
   - Implement recursive relations such as factorial and Fibonacci.
6. Search and Backtracking
   - Demonstrate Prolog's backtracking with sample queries and control cuts.
7. Family Relationship Programs
   - Model family trees and query relationships like siblings, ancestors.
8. Solve the Eight Queens Problem
   - Classic AI problem solved using backtracking.
9. Implement Simple Arithmetic Operations
   - Addition, subtraction, multiplication using Prolog predicates.
10. Monkey and Banana Problem
    - Logic problem modeling and solution.
11. Basic Expert System Prototype
    - Write rules for a simple medical diagnosis or recommendation system.
12. Implement Search Algorithms (Conceptual)
    - Demonstrate basic search algorithms like best-first search using Prolog rules.

SEMESTER-III

COURSE 3 : CONVERSATIONAL AI

Theory **Credits: 3** **3 hrs/week**

Course Objectives

1. Understand the fundamentals and significance of conversational AI and chatbots.
2. Learn to design conversation flows and manage dialogs effectively.
3. Gain hands-on experience in building and training Natural Language Understanding (NLU) models.
4. Develop skills for creating multilingual and voice-enabled chatbots.
5. Testing, deployment, and maintenance of conversational AI systems.

Unit 1: Introduction to Conversational AI and Chatbots What is Conversational AI and its importance, Overview of chatbot use cases and domains, Key components: NLU, dialog management, NLG, Dialog systems and architecture, Introduction to chatbot platforms and tools, Differences between rule-based and AI chatbots, Challenges in conversational AI development

Unit 2: Designing Conversations and Dialog Management Understanding conversation flows and user intents, Dialog management strategies: state machines, frame-based systems, Slot filling techniques for context maintenance, Handling multi-turn conversations and context switching, Managing conversation errors and fallback strategies, Role of session and user context in dialog personalization, Best practices in conversation design for UX

Unit 3: Natural Language Understanding (NLU) Fundamentals Preparing training annotation and labeling, intent classification techniques and algorithms , Entity recognition and extraction methods, Using pre-trained models and transfer learning in NLU, Handling ambiguous and overlapping intents, Supporting multiple languages with NLU components, Evaluating NLU model performance

Unit 4: Building Multilingual and Voice-Enabled Chatbots Techniques to support multilingual conversations, Speech recognition and synthesis basics , Integrating voice assistants with chatbots, text-to-speech (TTS) and speech-to-text (STT) , Designing voice user interfaces (VUIs), Testing voice chatbots with different accents and dialects, Privacy and ethical considerations in voice-enabled AI

Unit 5: Testing, Deployment, and Maintenance of Chatbots Testing chatbots: functional, user acceptance, load testing, Metrics for chatbot performance and user satisfaction, Continuous learning, retraining, and updating models, Handling failures and graceful degradation, Deploying chatbots across platforms and integration with backend systems, Analytics and monitoring tools for conversational AI, Future trends in conversational AI and emerging technologies

Recommended Textbooks & References

- "Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto, MIT Press
- "Conversational AI: Chatbots that work" by Andrew Freed, Packt Publishing
- "Handbook of Natural Language Processing" edited by Nitin Indurkhya and Fred J. Damerau
- "Conversational Interfaces: Principles and Practice" by Michael McTear
- "Deep Learning for Natural Language Processing" by Palash Goyal, Sumit Pandey, Karan Jain

SEMESTER- III

COURSE 3: CONVERSATIONAL AI

Practical	**Credits: 1**	**2 hrs/week**

Experiment 1: Setting Up Rasa

- Install Rasa and prerequisite tools
- Understand folder structure and key files (domain.yml, nlu.yml, stories.yml)
- Create a basic rule-based chatbot and test on Rasa shell

Experiment 2: Defining Intents and Entities

- Create intents and entities in nlu.yml
- Annotate training data for intent classification and entity extraction
- Train and test the NLU model with sample queries

Experiment 3: Dialogue Management with Stories and Rules

- Write stories for conversation flows (user intents + bot responses)
- Define rules for slot filling and fallback handling
- Test multi-turn conversations and context handling

Experiment 4: Implementing Slots and Form Actions

- Use slots to capture and remember user inputs
- Create forms to fill multiple slots during conversations
- Manage slot validity and resetting

Experiment 5: Custom Actions for Dynamic Responses

- Write custom Python actions (in actions.py) for dynamic content
- Use APIs or simple logic to enhance chatbot responses
- Test bot with custom actions integrated

Experiment 6: Multilingual Chatbot Development

- Extend chatbot to support multiple languages via separate NLU pipelines
- Handle intents and responses per language
- Test language switching in conversations

Experiment 7: Integrating Voice with Rasa

- Connect Rasa chatbot with speech-to-text and text-to-speech services
- Build a simple voice-enabled chatbot demo
- Evaluate voice input accuracy and bot response

SEMESTER-IV

COURSE 4 : ROBOTICS PRINCIPLES AND EMBEDDED SYSTEMS

Theory **Credits: 3** **3 hrs/week**

Course Objectives

1. To introduce fundamental concepts of robotics including robot classification, kinematics, and mechanisms.
2. To explain the roles and types of actuators, drives, and sensors, including piezoelectric technologies.
3. To develop an understanding of robot control systems, feedback mechanisms, and intelligent control methods.
4. To expose students to embedded systems, focusing on microcontrollers and interfacing with hardware components.
5. To provide practical experience in Arduino programming and hardware interfacing for robotics applications.

Course Outcomes

Upon successful completion of this course, students will be able to:

1. Classify different types of robots and explain their kinematic principles and degrees of freedom.
2. Describe actuators, drives, and sensor technologies used in robotics, including piezoelectric actuators and sensors.
3. Design and analyze control systems for robotic joints with feedback and intelligent control strategies.
4. Understand embedded systems concepts and program microcontroller-based systems (Arduino) for robotic applications.
5. Develop and implement practical robotics solutions using Arduino hardware and software tools.

Unit 1: Introduction to Robotics

Classification of Robots, Components and Characteristics of Robots, Applications of Robotics, Degrees of Freedom: Definition, importance, calculation, examples

Fundamentals of Kinematics and Position Analysis, Robots as Mechanisms, Matrix Representation and Transformation Matrices, Forward and Inverse Kinematics

Unit 2: Actuators and Drives

Actuating Systems: Features, selection criteria, **Actuators:** Types- electric, hydraulic, pneumatic, **Drives:** Types and characteristics-electric, hydraulic, pneumatic, stepper, servo; comparison and usage, **Piezoelectric actuators:** Principle, applications in microrobotics, advantages (precision, quick response, energy harvesting) , Use of Reduction Gears, Actuating Devices and Control

Unit 3: Grippers and Sensors

Grippers: Types (mechanical, magnetic, vacuum, adhesive), applications, Criteria for gripper selection, Sensor characteristics, Description of Vision, Force, Proximity, Tilt Sensors, Piezoelectric sensors (Working principle, parts, & applications), types (force, vibration, pressure), applications of robotics, Principles and practical applications of key sensor types

Unit 4: Robot Controls

Point-to-Point and Continuous Path Control, Intelligent Robot Control Methods, Control Systems for Robot Joints, Control Actions, **Feedback Devices:** Concepts and significance, Integration of sensors in feedback and control loops (e.g., vibration damping, tactile feedback, health monitoring), Closed-loop vs. Open-loop Control Systems

Unit 5: Embedded Systems

Embedded Systems: Definition, characteristics, and significance in robotics, Microcontrollers and Microprocessors, Arduino microcontroller architecture and applications, Interfacing sensors, actuators, drives, and grippers with embedded systems, Embedded programming essentials (C/C++, input/output, analog/digital signals), Embedded communication protocols (serial, I2C basics), Integration of sensor data in embedded applications.

Unit 1: Introduction to Robotics

Activity: Analyze and classify various types of robots; calculate degrees of freedom for given robot models.

Outcome: Students will understand robot types, components, and the significance of degrees of freedom.

Evaluation Method: Written assignments and quizzes on robot classification and kinematics concepts.

Unit 2: Actuators and Drives

Activity: Investigate characteristics of electric, hydraulic, pneumatic actuators and drives; study piezoelectric actuator demonstrations.

Outcome: Ability to compare different actuators and drives and understand piezoelectric actuator applications.

Evaluation Method: Lab demonstration reports and presentation on actuators.

Unit 3: Grippers and Sensors

Activity: Examine different types of grippers; conduct hands-on experience with piezoelectric, proximity, force, and vision sensors.

Outcome: Understand gripper types, sensor characteristics, and apply piezoelectric sensing principles.

Evaluation Method: Practical sensor interfacing reports and sensor data analysis.

Unit 4: Robot Controls

Activity: Implement point-to-point and continuous path control algorithms; integrate sensor feedback for closed-loop control.

Outcome: Develop understanding of control methods and feedback systems in robotics.

Evaluation Method: Lab assignments implementing control algorithms on robotic joints; tests on control theory.

Unit 5: Embedded Systems

Embedded Systems: Definition, characteristics, and significance in robotics, Microcontrollers and Microprocessors, Arduino microcontroller architecture and applications, Interfacing sensors, actuators, drives, and grippers with embedded systems, Embedded programming essentials (C/C++, input/output, analog/digital signals), Embedded communication protocols (serial, I2C basics), Integration of sensor data in embedded applications

Recommended Textbooks & References

1. "Robotics: Mechanics and Control" by Guruprasad K. Eastern Economy Edition
2. "Modern Robotics: Mechanics, Planning, and Control" by Kevin M. Lynch and Frank C. Park, Cambridge University Press.

3. “Introduction to Autonomous Mobile Robots” by Roland Siegwart et al. (Free e-Resource)
4. “Arduino Programming in 24 Hours” by Richard Blum, Sams Teach Yourself
5. “Springer Handbook of Robotics” edited by Bruno Siciliano and Oussama Khatib.

SEMESTER-IV

COURSE 4: ROBOTICS PRINCIPLES AND EMBEDDED SYSTEMS

Practical **Credits: 1** **2 hrs/week**

Note: This lab has to be completed using Arduino Micro controller and Arduino programming.

1. Introduction to Arduino and Basic Programming
   - Setting up Arduino IDE
   - Blinking an LED — basic digital output
2. Reading Digital Inputs
   - Interfacing and reading button presses/switches
3. Interfacing Analog Sensors
   - Reading values from a potentiometer or light-dependent resistor (LDR)
4. Ultrasonic Distance Sensor Interfacing
   - Measuring distance using HC-SR04 sensor and displaying data
5. Servo Motor Control
   - Controlling servo motor angles for robotic joint simulation
6. DC Motor and Motor Driver Interfacing
   - Running a DC motor with L298N motor driver
7. Stepper Motor Control
   - Basic control of stepper motors for precise movements
8. Line Following Robot Implementation
   - Using IR sensors to detect and follow a line on the ground
9. Obstacle Avoidance Robot
   - Using ultrasonic sensors and motor control to avoid obstacles
10. Force/Tactile Sensor Interfacing (including Piezo sensors)
    - Reading piezoelectric sensor signals as force or vibration inputs
11. Temperature Sensor Interfacing
    - Using sensors like LM35 or DHT11 for environmental sensing
12. Bluetooth Module (HC-05) Controlled Robot
    - Controlling robot movement wirelessly using smartphone commands
13. Servo-Driven Robotic Gripper
    - Building and controlling a simple gripper mechanism
14. PWM Motor Speed Control

Controlling motor speed using Pulse Width Modulation techniques

KRISHNAUNIVERSITY

Diploma (AI&Robotics) Examinations Model Paper for Core Subjects

Semester: I

Paper Title: PYTHON PROGRAMMING AND DATA STRUCTURES

Time: 3hrs **Max.Marks:70**

Section–A(20Marks)

Answer any *Five* questions. All questions carry equal marks. **5X4=20M**

1. Built-in Data Types in Python.
2. Explain about Input/Output Statements in Python?
3. Describe Functions in Python.
4. What are Variables?
5. Explain about Dictionaries?
6. Explain about Encapsulation.
7. Define Abstract Data Structures (ADTs)?
8. What is FIFO data structure?

Section–B(50Marks)

Answer ALL Questions. All questions carry equal marks. **5X10 =50M**

9. a) Explain about the Features of Python.

OR

b) Classification of operators in Python.

10. a) Explain about the Iterative Statements in Python.

OR

b) Explain about Python Modules.

11. a) Briefly explain about Tuples in Python.

OR

b) Write about Lists in Python.

12. a) Explain about Exception Handling.

OR

b) Briefly explain about Inheritance.

13. a) Briefly explain about Linked Lists.

OR

b) Explain about Stacks and Queues.

KRISHNAUNIVERSITY

Diploma (AI&Robotics) Examinations Model Paper for Core Subjects

Semester: II

Paper Title: ARTIFICIAL & COMPUTATIONAL INTELLIGENCE

Time: 3hrs **Max.Marks:70**

Section–A(20Marks)

Answer any *Five* questions. All questions carry equal marks. 5X4=20M

1. Applications of AI.
2. What are Intelligence Agents?
3. What are the Examples of Expert Systems?
4. What are Rule-based systems?
5. Explain about Traveler's problem?
6. Explain about Regression.
7. Define Responsible AI.
8. Role of computational intelligence in AI.

Section–B(50Marks)

Answer ALL Questions. All questions carry equal marks. 5X10 =50M

9. a) Explain about the PEAS framework.

OR

b) Discuss about Types of intelligent agents.

10. a) What is Expert System? Explain the components of Expert Systems

OR

b) Explain about Role and significance of expert systems in AI evolution.

11. a) Briefly explain about Uninformed (Blind) Search methods.

OR

b) Write about Informed (Heuristic) Search.

12. a) What is machine learning? Explain about the types of Machine Learning.

OR

b) What are the Basic Learning Algorithms in ML?

13. a) Overview of computational intelligence.

OR

b) What are the Ethics and societal challenges in AI?

KRISHNAUNIVERSITY

Diploma (AI&Robotics) Examinations Model Paper for Core Subjects

Semester: III

Paper Title: CONVERSATIONAL AI

Time: 3hrs **Max.Marks:70**

Section–A(20Marks)

Answer any *Five* questions. All questions carry equal marks. **5X4=20M**

1. Overview of chat bots.
2. What are Dialog systems?
3. What are the Conversations and Dialog Management?
4. Handling multi-turn conversations and context switching?
5. Explain about Entity recognition and extraction methods?
6. Privacy and ethical considerations in voice-enabled AI.
7. Future trends in conversational AI.
8. Maintenance of Chatbots.

Section–B(50Marks)

Answer ALL Questions. All questions carry equal marks. **5X10 =50M**

9. a) What are Conversational AI and its importance?

OR

b) Challenges in conversational AI development.

10. a) What are Dialog management strategies

OR

b) Explain about Managing conversation errors and fallback strategies.

11. a) Briefly explain about Natural Language Understanding (NLU) Fundamentals.

OR

b) Write about Using pre-trained models and transfer learning in NLU.

12. a) Building Multilingual and Voice-Enabled Chat bots Techniques.

OR

b) What are the text-to-speech (TTS) and speech-to-text (STT) , Designing voice user interfaces (VUIs)?

13. a) Overview of testing chat bots.

OR

b) What are the Analytics and monitoring tools for conversational AI?

KRISHNAUNIVERSITY

Diploma (AI&ML) Examinations Model Paper for Core Subjects

Semester: IV

Paper Title: ROBOTICS PRINCIPLES AND EMBEDDED SYSTEMS

Time: 3hrs **Max.Marks:70**

Section–A(20Marks)

Answer any *Five* questions. All questions carry equal marks. 5X4=20M

1. Classification of Robots.
2. What are Degrees of Freedom?
3. Describe piezoelectric actuators.
4. Use of Reduction Gears?
5. Write about Tilt Sensors?
6. Explain about Feedback Devices.
7. Define embedded systems?
8. Closed-loop vs. Open-loop Control Systems.

Section–B(50Marks)

Answer ALL Questions. All questions carry equal marks. 5X10 =50M

9. a) Explain about the Components and Characteristics of Robots, Applications of Robotics.

OR

b) Discuss about Fundamentals of Kinematics and Position Analysis.

10. a) Explain about the Actuators.

OR

b) Explain about Piezoelectric actuators.

11. a) What is Gripper? Briefly explain about Gripper Types.

OR

b) Write about Sensor Characteristics. Types of Sensors

12. a) Explain about Robot Control Methods.

OR

b) Integration of sensors in feedback and control loops.

13. a) Briefly explain about Microcontrollers and Microprocessors.

OR

b) Explain about Embedded communication protocols.