GOVT. M. H. COLLEGE OF HOME SCIENCE & SCIENCE FOR WOMEN (AUTONOMOUS), JABALPUR (M.P.)

COURSE OUTCOME CALENDAR

COMPUTER SCIENCE

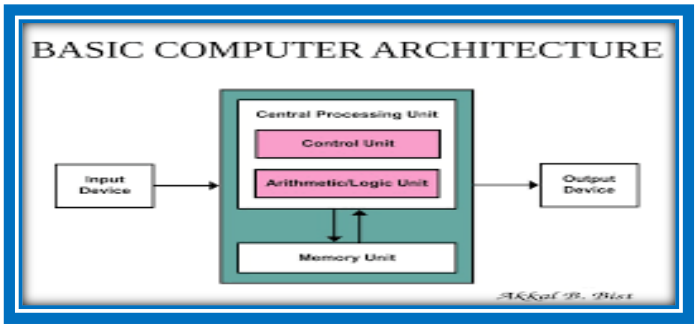
SESSION- 2024-25

Department of Mathematics & Computer

B.sc. I Year
(Computer Science)

Course Learning Outcomes

Paper-I (Course Code – S1-COSC1T (MAJOR))



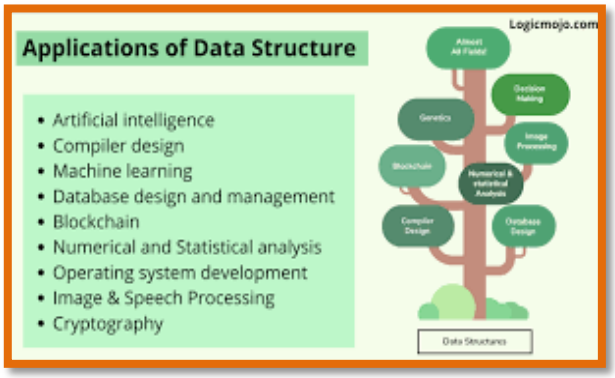
Course Title	Computer System Architecture

The course outcomes (COs) for a course on Computer System Architecture **typically focus on providing students with both theoretical and practical knowledge about how computer systems are designed, structured, and function.**

CO1:	**Understanding of Basic Computer Architecture Concepts:**	• **Describe the fundamental components of computer systems such as processors, memory, input/output devices, and storage.** • **Understand the function of the central processing unit (CPU), its organization, and execution of instructions**
CO2:	**Memory Hierarchy and Management**	**•Understanding the different types of memory (primary, secondary, cache, virtual memory) and how data is managed across these different layers.** • **Understanding how memory management techniques like paging, segmentation, and cache optimization work.**
CO3:	**Instruction Set Architecture (ISA) and Data Representation and Computer Arithmetic**	• **Understanding how instructions are formatted, decoded, and executed within a CPU.** • **Learning about different ISAs such as CISC (Complex Instruction Set Computing) and RISC (Reduced Instruction Set Computing).** • **Learning how data is represented in binary, hexadecimal, and other formats.** • **Understanding basic arithmetic operations on binary numbers, floating-point arithmetic, and overflow handling.**
CO4:	**Input/Output Systems&Parallelism and Multicore Architecture,Digital Electronics**	• **Understanding how data is transferred between the CPU and peripheral devices.** • **Learning about different I/O mechanisms like programmed I/O, interrupt-driven I/O, and direct memory access (DMA).** • **Understanding parallel processing and multicore processors.** • **Learning how multiple cores work together to handle simultaneous tasks and improve system performance.** • **Digital electronics is a branch of electronics that deals with digital signals and systems, as opposed to analog signals. I** • **In digital electronics, the information is represented in binary form (using two states, 0 and 1), typically through voltagesor other physical quantities.**
CO5:	**Logic Gates**	• **Logic gates perform logical operations on one or more binary inputs to produce a binary output.**

Paper-II Course Code – S1-COSC2T

(MAJOR/ MINOR/ ELECTIVE)



Course Title — Programming Methodologies & Data Structure

Programming Methodologies and Data Structures are two fundamental concepts in computer science and software development. They guide how programs are designed, structured, and implemented for efficiency and maintainability. Below is an overview of both:

1

Programming Methodologies

Some of the most common methodologies include:

- **Object-Oriented Programming (OOP)**
- **Functional Programming (FP)**
- **Declarative Programming**

2

Data Structures

Below are some common data structures:

a) Arrays b) Linked Lists c) Stacks d) Queues
e) Hash Tables f) Trees g) Heaps h) Graphs

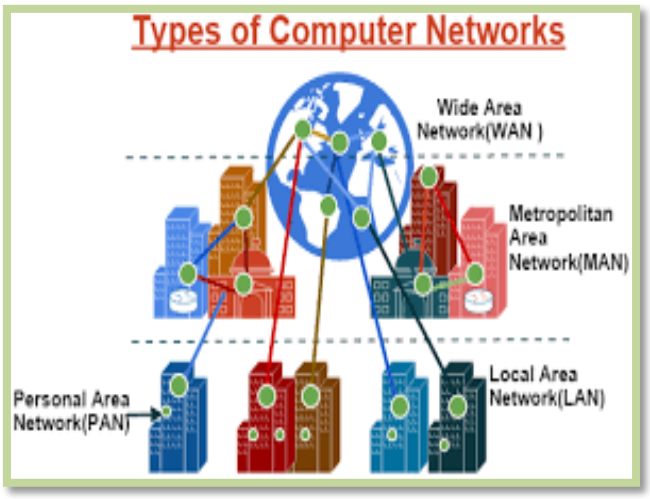
The course outcomes for Programming Methodologies & Data Structuretypically focus on equipping students with both theoretical and practical knowledge regarding efficient problem-solving, algorithm design, and the implementation of data structures.

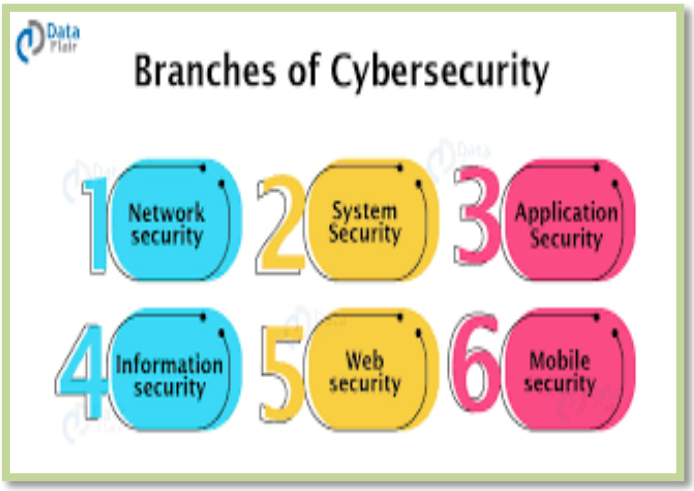
CO1:	**Understanding Programming Methodologies:**	• **Ability to choose and apply appropriate programming paradigms.** • **Problem-solving using algorithms.** • **Code Design and Structure.**
CO2:	**Understanding and Implementing Data Structures:**	• **Mastery of basic data structures** • **Application of advanced data structures.** • **Efficiency of data structures**
CO3:	**Algorithm Analysis and Design:**	• **Time and Space Complexity.** • **Sorting and Searching Algorithms.** • **Recursive Problem Solving.**
CO4:	**Problem-solving Skills:**	• **Debugging and Testing.** • **Application of Data Structures in Real-World Problems.**
CO5:	**Critical Thinking and Analytical Skills & Practical Experience**	• **Algorithm Optimization.** • **Complexity Analysis.**

B.sc. II Year
(Computer Science)

Course Learning Outcomes

Paper-I (Course Code – S2-COSC1T (MAJOR))



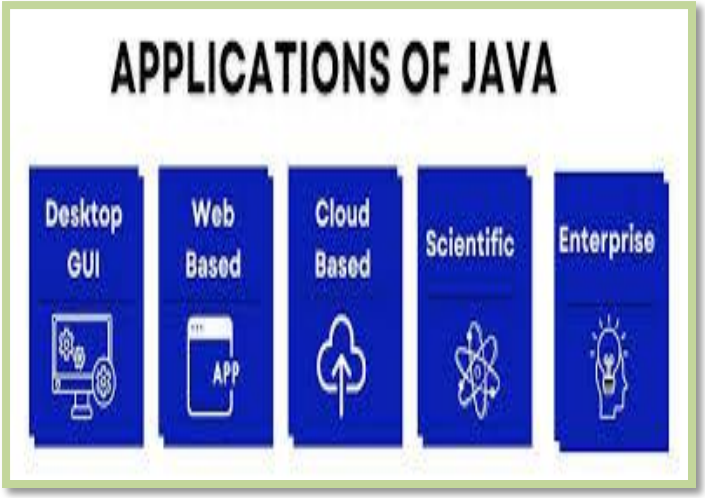


Course Title	Computer Networks & Information Security	
The course outcome for "Computer Networks & Information Security" typically aims to equip students with a strong foundation in the principles and technologies used in computer networks, along with the knowledge required to secure them.		
CO1:	**Understanding of Computer Networks**	**•Basic Networking Concepts •Network Architecture and Topology •Routing and Switching**
CO2:	**Reference Model**	**• A reference model in computer networking is a conceptual framework that standardizes the functions of a network into different layers. • OSI (Open Systems Interconnection) Model. • TCP/IP Model.**
CO3:	**Guided Transmission Medium:**	**• Twisted Pair Cable • Coaxial Cable. • Fiber Optic Cable.**
CO4:	**Switching Techniques**	**• Circuit Switching. • Packet Switching. • Message Switching.**
CO5:	**Ethical and Legal Aspects of Network Security,**	**•Ethical Hacking and Penetration Testing. •Cyber Laws and Regulations**

Course Learning Outcomes

Paper-II Course Code – S2-COSC2T

(MAJOR/ MINOR / ELECTIVE)T

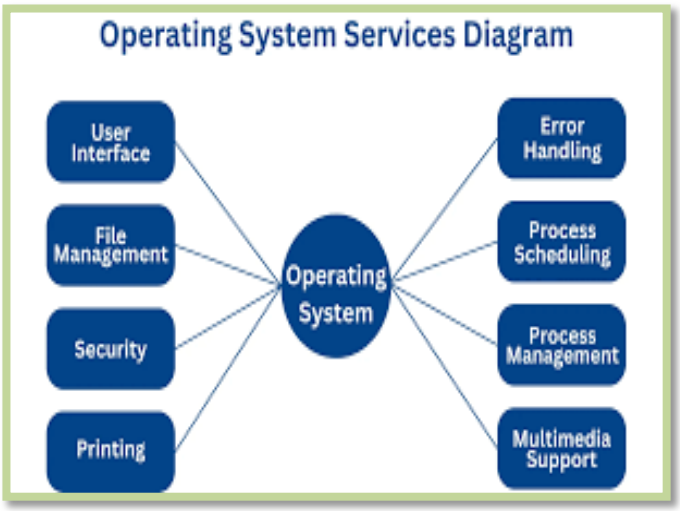


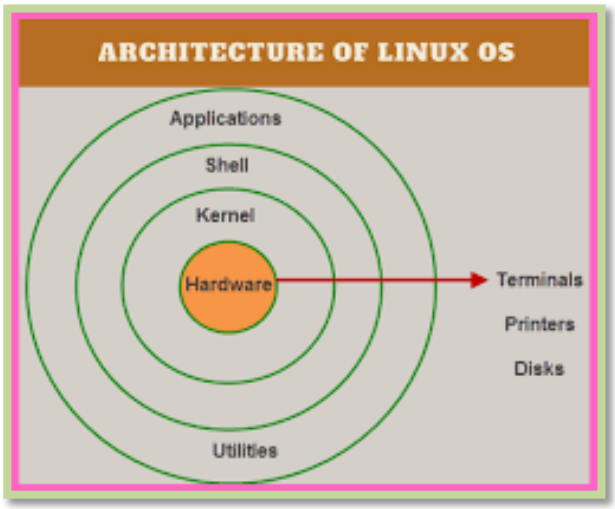
Course Title	Object Oriented Programming with Java	
The Course Outcome (CO) of an "Object-Oriented Programming with Java" course typically includes the following objectives and skills that students are expected to achieve by the end of the course:		
CO1:	Understand Object-Oriented Concepts:	• Grasp the foundational principles of Object-Oriented Programming (OOP), such as encapsulation, inheritance, polymorphism, and abstraction. • Demonstrate knowledge of Java's object-oriented features and how to apply them in real-world scenarios.
CO2:	Proficiency in Java Programming:	•Gain a strong understanding of Java syntax and semantics. • Write, debug, and execute Java programs using object-oriented techniques. • Use Java development tools (IDEs like IntelliJ IDEA, Eclipse, etc.) to effectively write and test code.
CO3:	Design and Implement Classes and Objects:	•Develop custom classes and create objects from those classes. • Design constructors, methods, and fields to implement class behaviors and attributes. • Utilize constructors for object initialization and understand constructor overloading.
CO4:	Exception Handling& Work with Java APIs	•Understand the exception hierarchy and handle exceptions using try, catch, and finally blocks. • Create custom exceptions to handle specific program conditions and improve program robustness. • Use common Java libraries and APIs such as java.lang, java.util, and java.io for building efficient and scalable applications.
CO5:	Applet tag	• An applet is a small Java program that runs in a web browser. • The <applet> tag allowed web developers to insert interactive content, such as games or dynamic features

B.sc. III Year
(Computer Science)

Course Learning Outcomes

Paper-I (Course Code – S3-COSC1D (MAJOR))





Course Title	Operating System

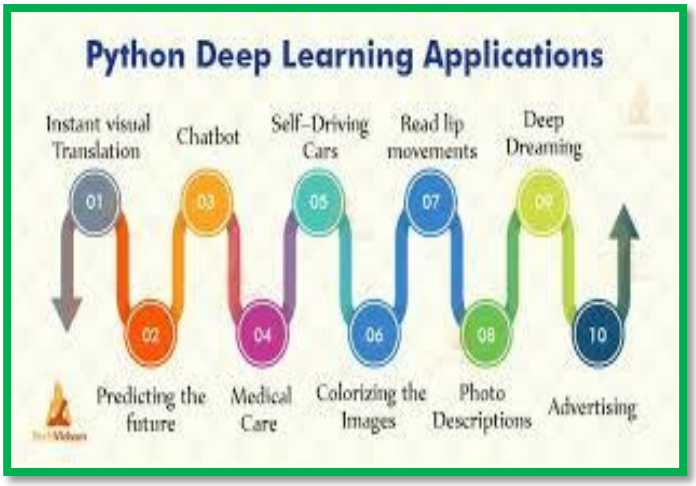
The course outcomes for an Operating System (OS) course typically focus on the fundamental concepts, principles, and techniques used in the design and implementation of modern operating systems.

CO1:	**Understand the Basics of Operating Systems:**	• **Define an operating system and explain its functions and services.** • **Understand the architecture and types of operating systems (e.g., batch, time-sharing, real-time, distributed).**
CO2:	**Process Management:**	• **Understand processes, process states, and process control.** • **Study process scheduling algorithms and analyze their performance.** • **Implement and solve problems related to process synchronization and inter-process communication (IPC).**
CO3:	**Memory Management:**	• **Understand the concept of memory management and memory hierarchy.** • **Learn about techniques such as paging, segmentation, and virtual memory.** • **Analyze memory allocation strategies and their trade-offs (e.g., contiguous vs. non-contiguous).**
CO4:	**File Systems & Input/Output (I/O) Management:**	• **Understand the components and operations of a file system.** • **Study file organization, file access methods, and file system implementation techniques.** • **Learn about disk scheduling algorithms and file system management.** • **Learn how the OS manages input/output devices and implements device drivers.** • **Study I/O scheduling techniques and buffering strategies.**
CO5:	**Understanding Linux Operating System:**	• **Gain a deep understanding of the architecture and components of the Linux operating system.** • **Learn the differences between Linux and other operating systems like Windows or Unix.**

Course Learning Outcomes

Paper-II Course Code – S3-COSC2D

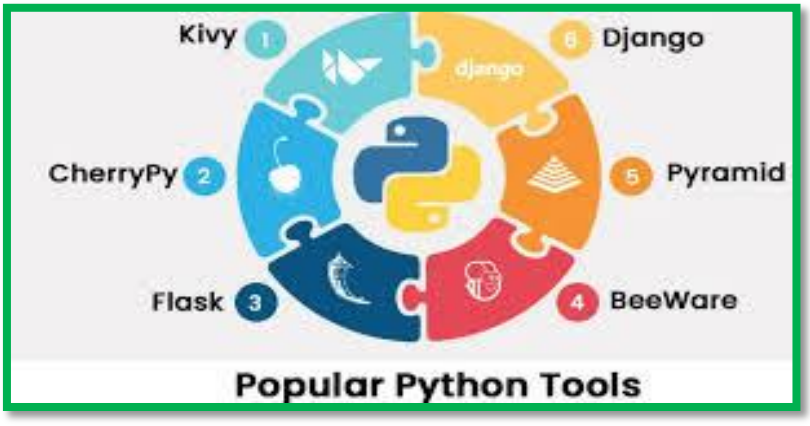
(MAJOR)



Course Title	Programming with Python

These outcomes are generic and can be adapted based on specific course objectives:

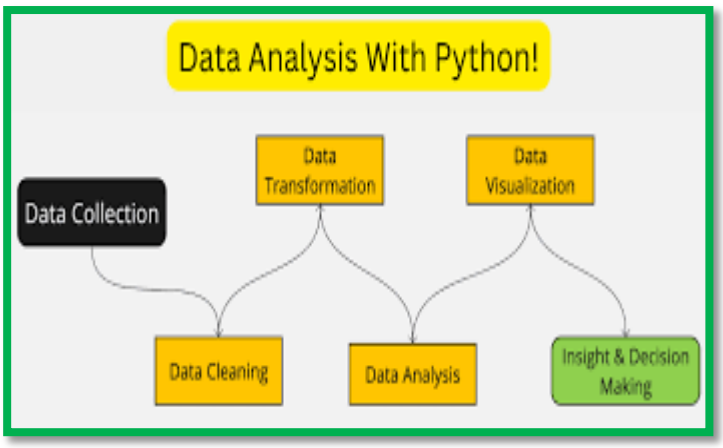
CO1:	**Understand Python Fundamentals**	• **Demonstrate proficiency in Python's syntax and semantics, including data types, operators, and control flow constructs.**
CO2:	**Develop Problem-Solving Skills**	• **Apply Python programming techniques to solve computational problems in a structured manner.**
CO3:	**Work with Data Structures**	• **Implement and manipulate essential data structures such as lists, tuples, dictionaries, and sets for effective data organization.**
CO4:	**Implement Functions and Modules , Handle Exceptions and Debugging**	• **Design modular programs using Python functions, modules, and packages to improve code reusability and maintainability.** • **Write robust programs by implementing error handling mechanisms and debugging tools to manage exceptions effectively.**
CO5:	**Perform File Handling Operations & Understand Basics of Databases**	• **Demonstrate the ability to read from, write to, and manipulate files for data storage and retrieval.** • **Connect Python programs to databases using libraries like SQLite to perform CRUD operations.**

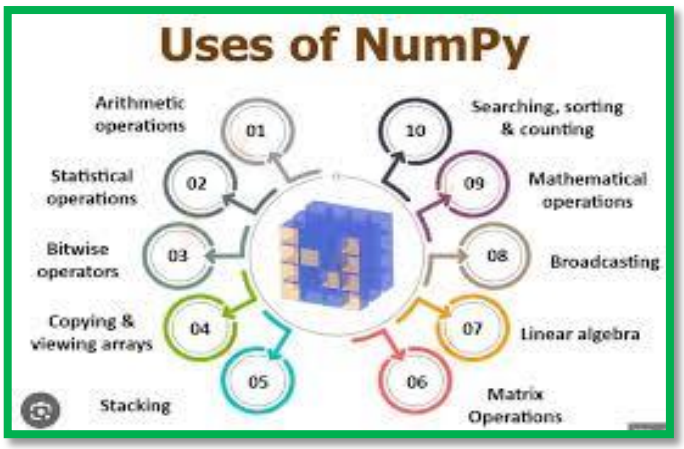


Course Learning Outcomes

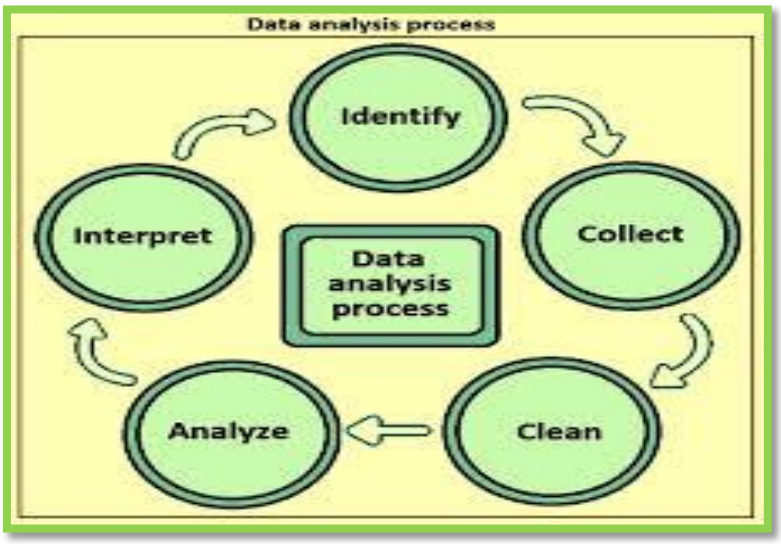
Course Code – S3-COSC2T

(MINOR / ELECTIVE)





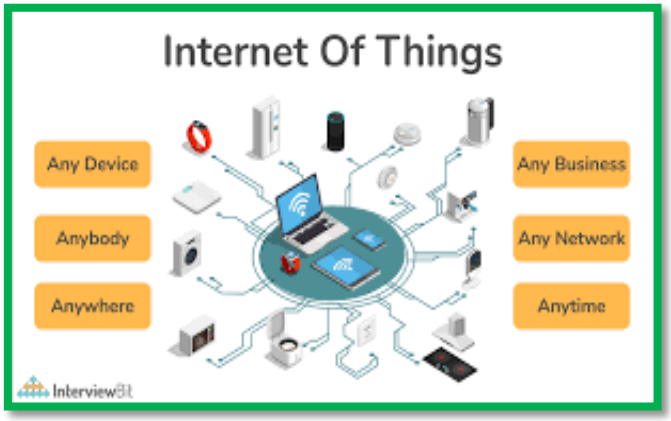
	Course Title	Data Analysis and Visualization with Python
CO1 :	**Understand Python Fundamentals**	• **Demonstrate proficiency in Python's syntax and semantics, including data types, operators, and control flow constructs.**
CO2 :	**Work with Data Structures**	• **Implement and manipulate essential data structures such as lists, tuples, dictionaries, and sets for effective data organization.**
CO3 :	**Understanding of NumPy Basics**	• **Gain foundational knowledge of NumPy, including its core concepts and the purpose of using NumPy in scientific computing and data analysis.**
CO4 :	**Efficient Array Manipulation**	• **Learn how to create, manipulate, and operate on NumPy arrays efficiently for numerical and data-oriented computations.**
CO5 :	**Data Visualization Skills**	• **Create effective and visually appealing data visualizations using libraries such as Matplotlib, Seaborn, and Plotly, tailored to specific audiences and purposes.**



B.sc. IV Year (Honours)
(Computer Science)

Course Learning Outcomes

Paper-CI (Course Code – S4-COSC1T (Core Course- C1)



Course Title		Internet of Things(IoT)
The course outcomes for an Internet of Things (IoT) course typically describe the knowledge, skills, and abilities students are expected to acquire upon completing the course.		
CO1:	**Understanding IoT Concepts and Architecture**	• **Describe the fundamental concepts, architecture, and protocols of the Internet of Things.**
CO2:	**IoT Communication and Networking**	• **Explain the communication models and networking standards used in IoT, including protocols like MQTT, CoAP, and HTTP.**
CO3:	**Basics of Arduino Programming**	• **Programming Language: Arduino uses a simplified version of C++.** • **Arduino IDE: The software where you write, compile, and upload code to an Arduino board.**
CO4:	**Understanding Python Fundamentals & ntegration of Python with Raspberry Pi**	• **Utilize Python to interact with Raspberry Pi's GPIO (General Purpose Input/Output) pins for controlling electronic components like LEDs, sensors, and motors.** • **Understand and implement IoT applications using Raspberry Pi, including connecting to the internet, sending/receiving data, and integrating with cloud platforms.**
CO5:	**Implement Fog Computing Solutions &**	• **Understanding Cloud Computing Concepts.** • **Develop an understanding of the key principles, theories, and concepts associated with FOG.** • **Analyze and evaluate various models or frameworks introduced in the course.** • **Apply theoretical knowledge to practical scenarios or case studies related to the subject.**

Course Learning Outcomes

Paper-C2 Course Code – S4-COSC2T
(Core Course-C2)

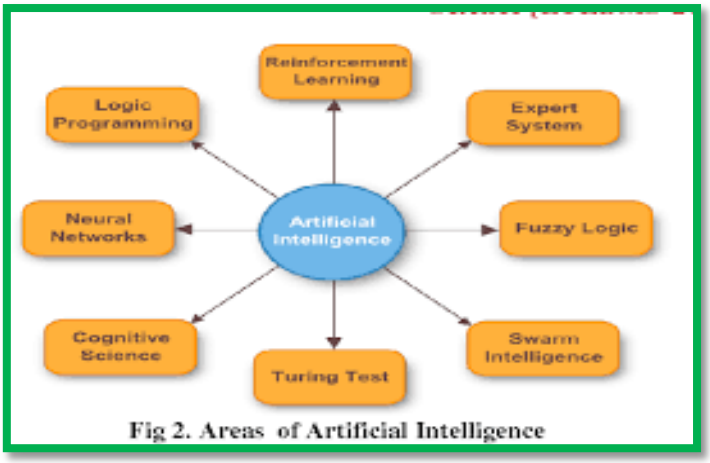
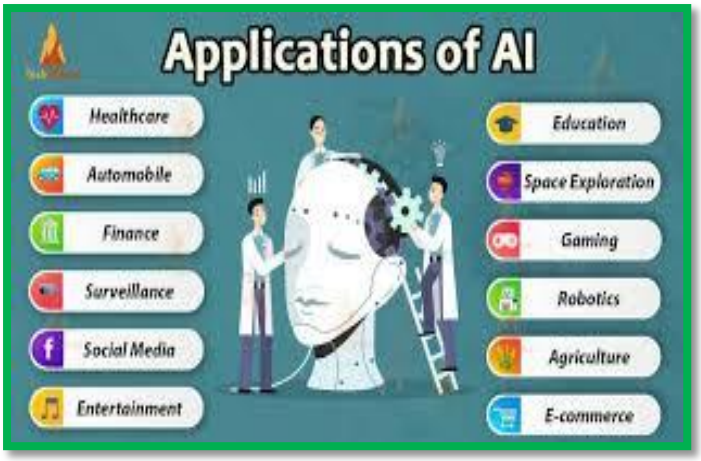


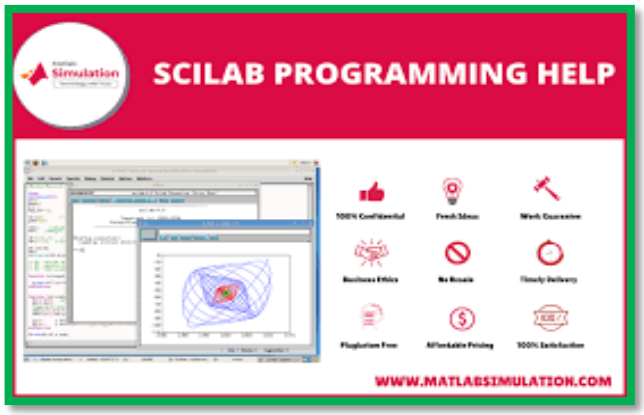
Fig 2. Areas of Artificial Intelligence



Course Title	Artificial Intelligence	
The course outcomes for an Artificial Intelligence (AI) course typically aim to ensure that students gain both theoretical understanding and practical skills in AI techniques and applications.		
CO1:	**Understanding AI Fundamentals:**	• **Gain a deep understanding of the core concepts and principles of Artificial Intelligence.** • **Learn about the history, evolution, and current trends in AI.** • **Understand the differences between strong and weak AI, and its applications in real-world problems.**
CO2:	**Proficiency in AI Search Algorithms**	• **Develop the ability to implement and evaluate various search algorithms such as Breadth-First Search (BFS), Depth-First Search (DFS), *A Search**, and Minimax Algorithm.** • **Understand heuristics and their role in improving search efficiency.**
CO3:	**Understand the Fundamentals of Knowledge Representation**	• **Comprehend the basic principles and theories of knowledge representation.** • **Understand various knowledge representation paradigms, including logic-based, frame-based, and semantic networks.**
CO4:	**Understanding HMMs Conceptually**	• **Students will develop a deep understanding of Hidden Markov Models, including their components, such as states, observations, transition probabilities, and emission probabilities. They will be able to explain how HMMs model time-series or sequence data.**
CO5:	**Understanding of Logic Programming & Proficiency in Prolog Syntax and Semantics:**	• **Students will be able to write and understand Prolog code, using its syntax for facts, rules, and queries.** • **They will learn how Prolog's underlying inference mechanism works, particularly backtracking, unification, and pattern matching.**

Course Learning Outcomes

Course Code – S4-COSC1D
(Discipline Specific Elective-D1)



Course Title		Computing with Scilab
The course outcomes (COs) for a typical Computing with Scilab course are designed to ensure students gain proficiency in using Scilab for numerical computing, programming, and problem-solving.		
CO1:	**Understanding of Scilab Environment**	• **Students will be able to effectively use the Scilab environment, including its basic commands, functions, and scripts.** • **They will gain an understanding of the interface, tools, and capabilities available in Scilab for numerical computation.**
CO2:	**Matrix and Linear Algebra Operations**	• **Students will be proficient in matrix manipulation, including operations like addition, multiplication, inversion, and finding eigenvalues and eigenvectors in Scilab.** • **They will apply these skills to solve linear algebra problems commonly encountered in engineering and scientific computing.**
CO3:	**Data Visualization**	• **Students will be able to generate and interpret 2D and 3D plots, graphs, and visualizations using Scilab's plotting tools.** • **They will learn to represent data and mathematical results visually to analyze and interpret them effectively.**
CO4:	**Polynomial Operations**	• **Perform basic operations on polynomials, including addition, subtraction, multiplication, and division.** • **Simplify polynomial expressions and combine like terms.** • **Understand and apply the distributive property to polynomials.**
CO5:	**Understanding of Scicos Environment:**	• **Gain a solid understanding of the Scicos graphical interface and its various features.** • **Familiarity with how to use Scicos to model and simulate dynamic systems, including continuous and discrete-time systems.**

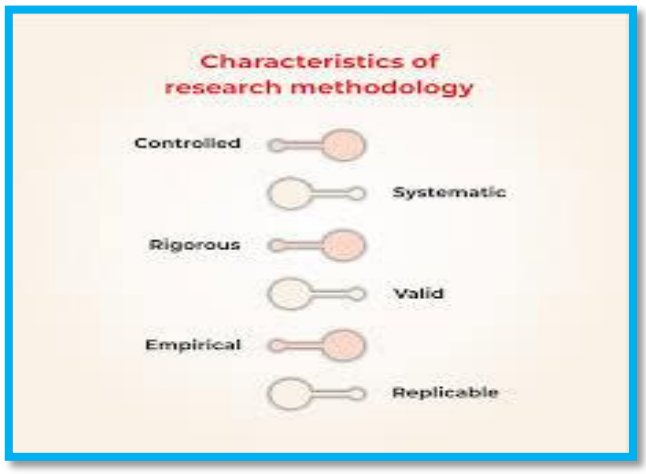
Course Learning Outcomes

Course Code – S4-COSC2Q
(Discipline Specific Elective-D2)

Course Title	Linux Server Adminstration	
The course outcomes for a Linux Server Administration course typically aim to equip students with the practical skills and knowledge necessary to manage and maintain Linux servers in a production environment.		
CO1:	Introduction to Linux Operating System	• Understand the basics of Linux architecture, including the kernel, shell, and file system structure. • Understand the differences between various Linux distributions (e.g., Ubuntu, • CentOS, Debian, RHEL).
CO2:	Linux Command-Line Proficiency	• Become proficient in using the Linux command line. • Understand file system navigation, permissions, and basic commands. • Use shell scripting for automating administrative tasks.
CO3:	Linux Installation and Configuration	• Install and configure different Linux distributions. • Set up partitioning, file systems, and disk management. • Configure network settings, hostname, and DNS on Linux systems.
CO4:	Understand the Linux Shell Environment & Write Shell Scripts: .	• Gain familiarity with the structure and functionality of different Linux shells (e.g., Bash, Zsh). • Understand basic shell features such as environment variables, file systems, permissions, and process management. • Develop shell scripts to automate common tasks such as file backups, system monitoring, and report generation. • Understand shell syntax, control flow (loops, conditionals), and how to handle input and output redirection.
CO5:	Linux Server Services Web and Database Server Management	• Configure common server services like DNS, DHCP, NFS, Samba, and email servers. • Implement file sharing, user authentication services, and printing services. • Install and configure Apache/Nginx for web server management. • Install and manage databases (e.g., MySQL, PostgreSQL) on a Linux server. • Set up web applications and ensure secure server configurations.

Course Learning Outcomes

Course Code – X4-AREM1T
(Research Methodology)



Course Title	Research Methodology

Course Outcome for Research Methodology typically focuses on equipping students with the essential skills and knowledge needed to effectively conduct research.

CO1:	**Understanding of Research Fundamentals:**	• **Students will gain a deep understanding of research concepts, principles, and Methodologies.** • **They will be able to distinguish between various types of research, such as qualitative, quantitative, exploratory, descriptive, experimental, and applied research.**
CO2:	**Development of Independent Research Skills:**	• **Students will develop the capability to conduct independent research from the problem identification phase to data collection and analysis.** • **They will gain confidence in managing their own research projects, including time management and resource allocation**
CO3:	**Research Design and Methodology:**	• **Students will understand the different types of research designs (e.g., experimental, descriptive, case study) and be able to select the most appropriate design for a given research question.** • **They will be able to design surveys, questionnaires, interviews, and experiments to collect data effectively**
CO4:	**Literature Review Skills:**	• **Students will gain the ability to conduct thorough literature reviews using academic databases and other resources.** • **They will understand how to critically analyze and synthesize existing research to identify gaps and build on previous knowledge.**
CO5:	**Formulating Research Problems and Hypotheses:**	• **Students will learn to identify and define research problems clearly.** • **They will develop skills in formulating research questions and hypotheses that guide the research process.**