GIRIJANANDA CHOWDHURY UNIVERSITY,ASSAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE STRUCTURE OF B.TECH 4TH SEMESTER

SL. NO.	COURSE CODE	COURSE NAME	L	T	P	C	CA	ESE
1	BCS23206T	DATA COMMUNICATION	3	0	0	3	40	60
2	BCS23207T	COMPUTER ORGANIZATION AND ARCHITECTURE	3	1	0	4	40	60
3	BCS23208T	OPERATING SYSTEM	3	0	0	3	40	60
4	BCS23209T	JAVA PROGRAMMING	3	0	0	3	40	60
5	BCS23209P	JAVA PROGRAMMING LAB	0	0	2	1	20	30
6	BCS23210T	GRAPH THEORY	3	0	0	3	40	60
7	BCS23211T	ADVANCED PYTHON PROGRAMMING	3	0	0	3	40	60
8	BCS23211P	ADVANCED PYTHON PROGRAMMING LAB	0	0	2	1	20	30
9	BEL23212T	IKS IV: Indian Science, Engineering & Technology (Past , Present & Future)	3	0	0	3	40	60
TOTAL CREDIT						24	320	480

	DATA COMMUNICATION (BCS23206T)	L	T	P	C
		3	0	0	3

Pre-requisite:

Basic mathematics, Basic electronics

Course Objectives:

1. To introduce basic concepts of data communication and networking.
2. To explore data transmission techniques including analog and digital communication, and different transmission media, guided and unguided.
3. To study different encoding, modulation and multiplexing techniques, along with error detection/correction techniques.
4. To learn standard LAN protocols and switching techniques.

Course Outcome:

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

CO1: **Understand** the data communication principles and networking basics.

CO2: **Evaluate** different analog and digital transmission techniques, including modulation, encoding, and multiplexing, and error detection/correction techniques.

CO3: **Analyze** transmission mediums used for communication, and different medium access control mechanisms.

CO4: **Understand** LAN protocols and switching techniques.

MODULE 1: Introduction — **5 hours**

ISO - OSI reference model; Protocol; Interface and service concepts; Layer wise functionality.

MODULE 2: Physical layer — **8 hours**

Concepts of data transmission; Signals: Analog, Digital: Signal properties; Periodic and Aperiodic Signals; Composite Signals; Time-domain and frequency-domain representations of signal; Channel capacity; Transmission impairments

MODULE 3: Encoding, Modulation, Multiplexing — **12 hours**

Line coding: NRZ, RZ, Manchester, Differential Manchester, Bipolar; Digitization: PCM, Delta modulation; Analog modulation: AM, FM, PM, Digital Modulation: ASK, FSK, PSK, QAM; Multiplexing: FDM, TDM, WDM, CDM, OFDM, Spread spectrum techniques.

MODULE 4: Transmission media — **7 hours**

Guided: Twisted pair, Coaxial cables, Optic fibers; Unguided: EM waves: Radio, Microwaves, Infrared; Satellite communication; Propagation models; SONET, Frame relay, ATM etc.

MODULE 5: Data link layer — **8 hours**

Types of Errors, Error Detection, Parity Check, Vertical Redundancy Check Longitudinal Redundancy Check, Cyclic Redundancy Check, Checksum, Error Correction; MAC protocols: ALOHA, CSMA, CSMA/CD, CSMA/CA, Token ring.

MODULE 6: Standard LAN protocols	**5 hours**
Network devices: Repeaters, Hub, Bridges, Switches, Routers, Gateways; Switching techniques: Circuit, Packet, Message; FDDI; VLAN; WLAN; PAN; WiMax.	
Total Lecture hours	**45 hours**

Text Book

(1) Forouzan B. A, Data Communication and Networking, 5e, Tata McGrawHill

(2) Stalling, Data and Computer Communication, 8e, PHI (EEE)

(3) Horak, R, Communication Systems and Networks, Wiley

Reference Books

(1) Tanenbaum A.S., Computer Network, 5e, PHI (EEE)

(2) Leon-Garcia, Widijaja, I., Communication, 5e, PHI (EEE)

(3) Kurose J. F, and Ross, K W, Computer Networking: A Top-Down Approach

	COMPUTER ORGANISATION AND ARCHITECTURE	L	T	P	C
		3	1	0	4

Pre-requisite: Basic Digital Logic Design, Fundamentals of Computer Systems

Course Objectives:

1. To understand the structure and functional organization of computers, including instruction sets, CPU, memory, and I/O subsystems.
2. To explore arithmetic operations and CPU design principles, such as control units, micro-operations, and microprogramming.
3. To analyze memory hierarchy, cache performance, virtual memory concepts, and data transfer mechanisms in I/O subsystems.

Course Outcome:

After successful completion of the course, the students will learn

1. CO1: To understand the basic functionality and operation of computer fundamentals including architecture and arithmetic functions.
2. CO2: To analyse CPU design principles, including control units, micro-operations, and microprogramming.
3. CO3: To analyse the memory organization, types of memory and memory management and interfacing between the CPU and peripherals.
4. CO4: To apply the concepts of parallel processing, pipelining and interprocessor communication.

MODULE 1 — **12hours**

Structure of Computer: Introduction to Computer Architecture, Functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Introduction to Instruction Set Types of ISA; RISC, CISC. Instruction execution cycle, RTL interpretation of instructions.
Computer Arithmetic : signed number representation, fixed and floating point representations, character representation. Computer arithmetic –integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication –shift-and add, Booth multiplier, carry save multiplier, etc. Division restoring and non-restoring techniques, floating point arithmetic.

Module 2 — **9 hours**

Introduction to x86 architecture and instruction set, Registers, addressing modes, instruction set. CPU control unit design: hardwired and micro-programmed design approaches, Micro-Program example, MICRO-OPERATIONS: Arithmetic Micro-Operations, Logic Micro-Operations.

Module 3 — **8hours**

Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards, Instruction Pipeline, Parallel Processing

Module 4 — **12hours**

MEMORY SYSTEM: Memory Hierarchy, Semiconductor Memories, RAM (Random Access Memory), Read Only Memory (ROM), Types of ROM, Cache Memory, cache size vs. block size, Performance considerations, Virtual memory, Paging, mapping functions, replacement algorithms.

Module 5	**4 hours**
Peripheral devices and their characteristics: Input-output subsystems, I/O interface, Programmed IO, Memory Mapped IO, Interrupt Driven IO, DMA. privileged and non-privileged instructions, software interrupts and exceptions.	
Total Lecture hours	**45 hours**

Text Book

1."Computer Organization and Design: The Hardware/Software Interface", 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.

2."Computer Organization and Embedded Systems", 6th Edition by Carl Hamacher, McGraw Hill Higher Education.

3."Computer Architecture and Organization", 3rd Edition by John P. Hayes, WCB/McGraw-Hill

Reference Books

1."Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.

2."Computer System Design and Architecture", 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education.

3. Computer System Architecture: By M. Morris Mano.

4. Structured Computer Organization: By Tanenbaum.

5. Computer Organization: By Stallings.

6. Computer Architecture and Organization: By Hayes.

7. Microprocessor Architecture, Programming, and Applications with the 8085 Ramesh S. Gaonkar Pub: Penram International.

Internal Member/BoS External Member/BoS

BCS23208T	Operating System	L	T	P	C
		3	0	0	3

Prerequisite:

1. Knowledge of data structures, algorithms, and computer architecture.
2. Proficiency in programming languages like C or C++.

Course Objectives:

1. To make aware of different types of Operating System and their services.
2. To provide in depth knowledge on different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
3. To familiarize students deadlocks prevention and detection techniques
4. To provide various mechanisms involved in memory management techniques and managing virtual memory.
5. To familiarize students with the disk management functions and techniques

Course Outcome:

After Successful completion of the course, students will be able to

CO1: Apply concepts of operating system, process.
CO2: Analyse the concepts of process synchronization, inter process communication and Deadlock
CO3: Apply Memory management and virtual memory concept.
CO4: Analyse I/O Management, File Systems Management, Disk Management, Protection and Security Mechanism.

Module: 1	**8 hours**

Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls

Module: 2	**9 hours**

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads. Process creation system call.
Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive

Module: 3	**7 hours**

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

Module: 4	**8 hours**

Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention,

Detection and Avoidance, Banker's Algorithm, Recovery from Deadlock.	
Module: 5	**8 hours**
Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation –Hardware support for paging, Protection and sharing, Disadvantages of paging. Virtual Memory: Basics of Virtual Memory – Hardware and control structures –Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, first in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU)	
Module: 6	**5 hours**
I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software File Management: File Concept, Access Methods, Directory Structure, File System Structure, Allocation Methods, and Free-Space Management. Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, RAID Structure	
Total hours	**45 hours**

Text Book

1.	Operating System Concepts Essentials, 9th Edition by Avi Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2.	Operating Systems: Internals and Design Principles, 5th Edition, William Stallings, Prentice Hall of India.
3.	Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
4.	Operating Systems: A Modern Perspective, 2nd Edition by Gary J. Nutt, Addison-Wesley
5.	Design of the UNIX Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
6.	Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

Reference Books

1.	Abraham Silberschatz, Peter Baer Galvin, Greg Gagne (2006), Operating System Principles, 7th edition, Wiley India Private Limited, New Delhi.
2.	Deitel & Deitel (2008), Operating systems, 3rd edition, Pearson Education, India.
3.	Modern Operating Systems, Tanenbaum, Fourth Edition. Pearson Publication.
4.	Operating Systems, Internals and Design Principles, Stallings, Seventh Edition, Pearson Publication.

GIRIJANANDA CHOWDHURY UNIVERSITY
Hathkhowapara, Azara, Guwahati - 781017, Assam

BCS23209T	Java Programming	L	T	P	C
		3	0	0	3

Pre-requisite: Concept of OOP using C++

Course Objectives:

1. Apply the concepts of Java to solve problems.
2. Understand the principles of inheritance and polymorphism.
3. Implementation of packages and interfaces.
4. Understand the concepts of exception handling and multithreading.
5. Implementation of the design of Graphical User Interface using applets and swing controls.
6. Understand the concepts of JDBC, JSP.

Course Outcomes: After successful completion of the course, the students will learn:

CO1: **Understand** the foundational concepts of Java, including data types, variables, control structures, object-oriented principles, and the execution flow of programs. **(Cognitive Level: Understanding)**

CO2: **Apply** object-oriented programming concepts such as inheritance, polymorphism, interfaces, and exception handling to develop modular and robust Java applications. **(Cognitive Level: Applying)**

CO3: **Design** interactive GUI-based applications using AWT and Swing components, incorporating event handling and layout management. **(Cognitive Level: Creating)**

CO4: **Utilize** JDBC and JSP to connect Java applications with databases and create dynamic, web-based solutions. **(Cognitive Level: Applying)**

Module 1:	**10 hours**
History of java, features of java, data types, variables, static variables, constants, scope and lifetime of variables, operators, structure of java program with example, execution flow of java program, new keyword , type conversion and casting, conditional statements, loops, break and continue statements, arrays, class, object and its methods, constructors and its types, static fields and methods, access control, this reference, final keyword, overloading methods and constructors, recursion, garbage collection, JVM, JRE.	
Module 2:	**10 hours**
Inheritance: Inheritance types, super keyword, final classes and methods. Polymorphism: Method overloading and overriding, abstract classes and methods. Interfaces: Interfaces vs. abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface. Packages: Defining, creating and accessing a package, importing packages.	

Module 3: Exception handling: Define exception, advantages of exception handling, the classification of exceptions- exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, creating own exception sub classes. Multithreading: Define Thread, multithreading, thread life cycle, creating threads, interrupting threads, thread priorities, synchronizing threads, inter-thread communication, and producer consumer problem.	**8 hours**
Module 4: Applets – Define applets, differences between applets and applications, Life cycle of an applet, Passing parameters to applets. GUI Programming with Java- The AWT class hierarchy, Introduction to Swing, Swing vs. AWT, Hierarchy for Swing components, Overview of some Swing components – Jbutton, JLabel, JTextField, JTextArea, simple Swing applications, Layout management – Layout manager types – border, grid and flow Event Handling- Events, Event sources, Event classes, Event Listeners, Event sources and Listeners, Delegation event model, Examples: Handling a button click, Handling Mouse events, Adapter classes.	**10 hours**
Module 5: Components and architecture of JDBC, Types of JDBC Architecture, Working of JDBC, JDBC Drivers, Establishing JDBC Connection in Java, Types of Statements in JDBC. JSP architecture, Life cycle of JSP, Difference between Servlet and JSP.	**10 hours**
Total Lecture hours	**48 hours**

Text Books/ Reference Books:

1. Java Fundamentals – A Comprehensive Introduction, Herbert Schildt and Dale Skrien, TMH.

2. Core Java: An Integrated Approach – Dr R Nageswara Rao.

3. Java for Programmers, P.J.Deitel and H.M.Deitel, PEA (or) Java: How to Program, P.J.Deitel and H.M.Deitel, PHI.

4. Object Oriented Programming through Java, P. Radha Krishna, Universities Press.

5. Thinking in Java, Bruce Eckel, PE 4. Programming in Java, S. Malhotra and S. Choudhary, Oxford Universities Press.

GIRIJANANDA CHOWDHURY UNIVERSITY
Hathkhowapara, Azara, Guwahati - 781017, Assam

BCS23209L	Java Programming Lab	L	T	P	C
		0	0	2	1

Pre-requisite: Concept of OOP using C++

Course Objectives:

1. Apply the concepts of Java to solve problems.
2. Understand the principles of inheritance and polymorphism.
3. Implementation of packages and interfaces.
4. Understand the concepts of exception handling and multithreading.
5. Implementation of the design of Graphical User Interface using applets and swing controls.
6. Understand the concepts of JDBC, JSP.

Course Outcomes: After successful completion of the course, the students will learn:

CO1: **Develop** Java programs using fundamental and object-oriented concepts such as classes, objects, inheritance, and interfaces to solve real-world problems. (Cognitive Level: Creating)
CO2: **Apply** exception handling and multithreading techniques to create robust and efficient Java applications. (Cognitive Level: Applying)
CO3: **Design** graphical user interfaces (GUI) with AWT and Swing, integrating event-handling mechanisms and layout management. (Cognitive Level: Creating)
CO4: **Construct** database-driven and dynamic web-based applications using JDBC and JSP. (Cognitive Level: Creating)

Module 1: Basics of Java Programming	**6 hours**
1. Write a Java program to demonstrate the usage of data types, variables, constants, and type casting. 2. Implement a program to illustrate conditional statements (if, switch) and loops (for, while, do-while). 3. Develop a program to perform array operations such as insertion, deletion, and searching. 4. Create a Java program to define a class, create objects, and demonstrate the usage of constructors and this keyword. 5. Write a program to demonstrate method overloading and constructor overloading.	
Module 2: Object-Oriented Programming Concepts	**6 hours**
6. Implement single, multilevel and hierarchical inheritance in Java. 7. Write a program to demonstrate the usage of super and final keywords in inheritance. 8. Develop a program to illustrate method overriding. 9. Create a Java program to define interfaces, implement them, and access them through interface references.	

10. Demonstrate the creation and usage of packages and importing user-defined packages.	
Module 3: Exception Handling and Multithreading 11. Write a program to handle checked and unchecked exceptions using try, catch, throw, throws, and finally. 12. Develop a user-defined exception class and use it in a program. 13. Create a multithreaded program to demonstrate thread life cycle and priorities. 14. Implement thread synchronization and inter-thread communication (e.g., Producer-Consumer problem).	**6 hours**
Module 4: GUI Programming and Event Handling 15. Create a Java applet to demonstrate the applet life cycle and parameter passing. 16. Design a GUI application using AWT to handle basic events like button clicks. 17. Develop a Swing-based application to demonstrate the use of JButton, JLabel, JTextField, and JTextArea. 18. Implement layout management using BorderLayout, GridLayout, and FlowLayout. 19. Write a program to handle mouse and keyboard events using adapter classes.	**6 hours**
Module 5: Database and Web Programming 20. Write a Java program to connect to a database using JDBC. 21. Develop a program to use different types of JDBC statements (Statement, PreparedStatement, CallableStatement). 22. Create a simple dynamic web application using JSP to display and process user inputs.	**6 hours**
Total Lecture hours	**30 hours**

	GRAPH THEORY	L	T	P	C
		3	0	0	3

Pre-requisite:

Basic mathematics

Course Objectives:

1. To introduce basic concepts of graph theory and their various properties.
2. To study the structural properties of different types of graphs, including trees, planar graphs, and bipartite graphs.
3. To develop problem-solving skills using graph algorithms like BFS, DFS, shortest path algorithms, and spanning tree algorithms.
4. To explore advanced graph theory topics like graph coloring, matching, and network flows, and examine their applications in optimization and resource allocation.

Course Outcome:

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

CO1: **Understand** the basics concepts of graph theory.

CO2: **Analyze** different types of graphs based on their properties and real-world applications.

CO3: **Apply** graph algorithms to solve practical problems such as computing shortest paths, minimum spanning trees, and optimized network flows.

CO4: **Evaluate** real life problems by applying the concepts of graph planarity and matching.

MODULE 1: Introduction — **5 hours**

Graph: terminologies; Incidence and degree; Handshaking Lemma; Types of graphs; Complete, regular, bipartite graphs with their basic properties; Distance in graphs: length, eccentricity, diameter, radius; Graph operations; Isomorphism; Subgraphs and Union of graphs; Adjacency matrix and Incidence matrix of graph.

MODULE 2: Walks, Paths, Circuits — **9 hours**

Walks, Paths, Circuits, Cycles; Components and Connectedness algorithms; Eulerian graphs: properties and theorems; Fleury's algorithm and Chinese postman problem; Hamiltonian graph - necessary and sufficient conditions; Labeled trees: Cayley's theorem; Directed graph

MODULE 3: Trees and Connectivity — **10 hours**

Properties of trees; Pedant vertices in a tree; Center of a tree; Rooted binary trees; Spanning trees and forest, Minimum spanning tree algorithms: Prim's and Kruskal's; Fundamental circuits; Spanning trees of a weighted graph; Menger's theorem

Connectivity: Cut-sets and Cut-vertices; Bridges and blocks; Fundamental cut-sets; auto Orphism groups; Connectivity and separativity; Network flow; Max-flow min-cut theorem; Reconstruction problem

MODULE 4: Planar Graph — **7 hours**

Combinatorial and geometric dual; Kuratowski's graph; Euler's theorem; Detection of

planarity; Thickness and Crossings.	
MODULE 5: Graph Coloring	**5 hours**
Chromatic number: Chromatic polynomial; Vertex and Edge coloring**;** The four colour problem.	
MODULE 6: Matching and Cover	**4 hours**
Matching in bipartite graph; Maximal matching; Vertex cover and edge cover; Decomposition and domination	
MODULE 7: Graph Algorithms	**5 hours**
Breadth-First Search (BFS); Depth-First Search (DFS); Dijkstra's Algorithm for shortest paths; Bellman-Ford Algorithm; Floyd-Warshall Algorithm for all-pairs shortest paths.	
Total Lecture hours	**45 hours**

Text Book

(1) J. A. Bondy and U. S. R. Murthy, Graph Theory, Springer Verlag

(2) Douglas B. West. Introduction to Graph Theory, PHI

(3) N. Deo, Graph Theory with Applications to Engineering and Computer Science.

(4) R. Diestel, Graph Theory, Springer Verlag

Reference Books

(1) G. Chatrand, and O.R. Ollermann, Applied and Algorithmic Graph theory, McGraw Hills.

(2) B. Bollobas, Modern Graph Theory, Springer

	ADVANCED PYTHON PROGRAMMING	L	T	P	C
		3	0	0	3

Pre-requisite: Basic knowledge of python programming

Course Objectives: By the end of this course, students will be able to

1. Understand and apply advanced Python features such as assertions, decorators, generators, iterators, and multi-threading using the threading module.
2. Develop the ability to perform database operations in Python using SQLite, including executing and processing DDL and DML queries.
3. Build interactive GUI applications in Python using Tkinter components like buttons, labels, menus, and message boxes.
4. Gain practical skills in image processing, web scraping with Beautiful Soup, and performing data analysis and visualization using Pandas and Matplotlib.

Course Outcome:

After successful completion of the course, the students will learn

CO1: Understand and explain advanced Python concepts such as assertions, decorators, generators, iterators, and multi-threading.
CO2: Apply Python for database connectivity and execute DDL/DML queries with SQLite
CO3: Design and implement interactive GUIs using Tkinter components.
CO4: Analyze data with Pandas, visualize insights with Matplotlib, and perform basic image processing and web scraping.

MODULE 1: Advanced Python Constructs and Multi-threading	**10hours**

Assertion, Decorators, Generators, Iterators, Creation, and Execution of threads using threading module

Module 2: Database programming using Python	**6 hours**

Connecting to a database (SQLite) using Python, Sending DML and DDL queries and processing the result from a Python Program

Module 3: GUI in Python	**6hours**

Button, Canvas, Checkbutton, Entry, Frame, Label, Listbox, Menubutton, Menu, Message, Radiobutton, Scale, Scrollbar, Text, Toplevel, Spinbox, PanedWindow, LabelFrame, tkMessageBox

Module 4: Basic image processing and web scrapping using Python	**6 hours**

Introduction to digital image processing, Basic operations on an image, Crop, Scale, Rotate, Flip, Changing contrast, brightness and color, Edge detection, blur, sharpening, Beautiful Soup

Module 5: Data analysis and data visualization	**10 hours**

Introduction to Pandas, Pandas data structures – Series and DataFrame, Data wrangling using pandas, Loading a dataset into a dataframe o Selecting Columns from a dataframe, Selecting Rows from a dataframe, Adding new data in a dataframe, Deleting data from a dataframe Introduction to Matplotlib, Scatter plot, Line plot, Bar chart, Histogram, Box plot.	
Total Lecture hours	**38hours**

Text Book

1. Beazley, D., & Jones, B. K. (2013). **Python cookbook: Recipes for mastering Python 3** (3rd ed.). O'Reilly Media.
2. Chollet, F. (2021). **Deep learning with Python** (2nd ed.). Manning Publications.
3. Grinberg, M. (2018). **Flask web development: Developing web applications with Python** (2nd ed.). O'Reilly Media

4. McKinney, W. (2022). **Python for data analysis: Data wrangling with Pandas, NumPy, and Jupyter** (3rd ed.). O'Reilly Media.

5. Ramalho, L. (2022). **Fluent Python: Clear, concise, and effective programming** (2nd ed.). O'Reilly Media

Reference Books

1. Lutz, M. (2013). **Programming Python** (4th ed.). O'Reilly Media.
2. Mitchell, R. (2018). **Web scraping with Python: Collecting data from the modern web** (2nd ed.). O'Reilly Media

3. Slatkin, B. (2020). **Effective Python: 90 specific ways to write better Python** (2nd ed.). Addison-Wesley Professional.

4. Sweigart, A. (2020). **Automate the boring stuff with Python: Practical programming for total beginners** (2nd ed.). No Starch Press.

5. Vincent, W. S. (2022). **Django for professionals: Production websites with Python & Django** (3rd ed.). WelcomeToCode

6. Python Software Foundation. (n.d.). **Python 3 documentation**. Retrieved from https://docs.python.org/3/

	ADVANCED PYTHON PROGRAMMING LAB	L	T	P	C
		-	-	-	-

Pre-requisite: Basic knowledge of python programming

Course Objectives: By the end of this course, students will be able to

1. Develop Advanced Python Programs.
2. Implement Data Handling and GUI-Based Applications
3. Perform Image Processing and Web Scraping
4. Analyze and Visualize Data

Programs:

1. Write a Python program that uses assertions to ensure a user-provided number is positive.
2. Create a decorator that measures and prints the execution time of functions.
3. Implement a generator to yield Fibonacci numbers up to a specified limit.
4. Create a custom iterator for iterating over a list of prime numbers.
5. Write a program to demonstrate multi-threading where each thread calculates the square of a number from a list.
6. Create a program where threads access shared data safely using threading.Lock.
7. Write a program using SQLite Database to store student records. Insert, retrieve, update, delete records and execute query to calculate and display the average grade of students.
8. Design a GUI with buttons that change the text of a label on a button click.
9. Create a form using Entry widgets to accept user input (e.g., name, age) and display it in a label.
10. Build an application to display a list of cities in a Listbox with a Scrollbar.
11. Design a menu-driven application with options like "About" and "Exit," using tkMessageBox to display messages.
12. Write a program to perform cropping, scaling, rotation, and flipping of an image using the Pillow library. And adjust the brightness, contrast, and apply sharpening filters to an image.
13. Scrape the latest news headlines from a website using the BeautifulSoup library and save them to a text file. Also scrape table data from a webpage and store it in a CSV file.
14. Load a dataset into a Pandas DataFrame, select specific columns, filter rows, and add new columns. Perform group-by operations and calculate aggregated metrics such as sum and mean for a dataset.
15. Use matplotlib to visualize relationships in a dataset using scatter plots and line plots. Create histograms and box plots to display data distribution and outliers. Build a bar chart with custom titles, labels, legends, and colors.

Text Book

1. Beazley, D., & Jones, B. K. (2013). **Python cookbook: Recipes for mastering Python 3** (3rd ed.). O'Reilly Media.

2. Chollet, F. (2021). **Deep learning with Python** (2nd ed.). Manning Publications.

3. Grinberg, M. (2018). **Flask web development: Developing web applications with Python** (2nd ed.). O'Reilly Media

4. McKinney, W. (2022). **Python for data analysis: Data wrangling with Pandas, NumPy, and Jupyter** (3rd ed.). O'Reilly Media.

5. Ramalho, L. (2022). **Fluent Python: Clear, concise, and effective programming** (2nd ed.). O'Reilly Media

Reference Books

1. Lutz, M. (2013). **Programming Python** (4th ed.). O'Reilly Media.
2. Mitchell, R. (2018). **Web scraping with Python: Collecting data from the modern web** (2nd ed.). O'Reilly Media

3. Slatkin, B. (2020). **Effective Python: 90 specific ways to write better Python** (2nd ed.). Addison-Wesley Professional.

4. Sweigart, A. (2020). **Automate the boring stuff with Python: Practical programming for total beginners** (2nd ed.). No Starch Press.

5. Vincent, W. S. (2022). **Django for professionals: Production websites with Python & Django** (3rd ed.). WelcomeToCode

6. Python Software Foundation. (n.d.). **Python 3 documentation**. Retrieved from https://docs.python.org/3/

Internal Member/BoS

External Member/BoS

	Indian Science, Engineering and Technology (Past, Present and Future)	L	T	P	C
		3	0	0	3

Pre-requisite:

1. Basic of Mathematics.

Course Objectives:

1. To familiarize learners with major sequential development in Indian science, engineering and technology.
2. To review & strengthen the ancient discovery and research in physics, chemistry, maths, metallurgy, astronomy, architecture, textile, transport, agriculture and Ayurveda etc.
3. To help students to trace, identify and develop the ancient knowledge systems to make meaningful contribution to development of science today.
4. To help to understand the apparently rational, verifiable and universal solution from ancient Indian knowledge system for the scientific, technological and holistic development of physical, mental and spiritual wellbeing.

MODULE 1: Indian Traditional Knowledge; Science and Practices — **8 hours**

Introduction to the Science and way of doing science and research in India, Ancient Science in Intra & Inter Culture Dialogue & coevolution. Traditional agricultural practices, Traditional water-harvesting practices, Traditional Livestock and veterinary Sciences Traditional Houses & villages, Traditional Forecasting, Traditional Ayurveda & plant based medicine, Traditional writing Technology

MODULE 2: Ancient Indian Science (Physics, Chemistry, Maths) — **9 hours**

Physics in India: Vaisheshika darshan Atomic theory & law of motion, theory of panchmahabhoota, Brihath Shathaka (divisions of the time, unit of distance), bhaskarachaya (theory of gravity, surya siddhanta & sidhanta shriomani), Lilavati (gurutvakashan Shakti). Chemistry in India Vatsyayana, Nagarjuna,Khanda, Al-Biruni, Vagbhaṭa –building of the ras-shala (laboratory), working arrangements of ras-shala, material and equipment, Yaśodhara Bhaṭṭa-process of distillation, apparatus, saranasamskara, saranataila Mathematics in India: Baudhayana's Sulbasutras, Aryabhaṭa, Bhaskaracharya-I, Severus Sebokht, Syria, Brahmagupta, Bhaskaracharya-II, Jyeṣṭhadeva

MODULE 3: Ancient Indian Science (metallurgy, Astronomy, Architecture) — **9 hours**

Metallurgy in India: Survarṇa(gold) and its different types, prosperities, Rajata(silver), Tamra(copper), Loha(iron), Vanga(tin), Naga / sisa(lead), Pittala(brass) Astronomy in India Vedang Jyotish, aryabhatta siddhanta, Mahabhaskriya, Laghubhaskariya, vatesvarasiddhanta, Sisyadhivrddhida, Grahashyay, Goladhyaya, Karabakutuhala (Aryabhaṭa, Varahamihira, Brahmagupta, Vaṭesvara, Bhaskara, Paramesvara, NilakaṇṭhaSomayaji, Jyeṣṭhadeva, ŚankaraVarman) Architecture in India: Nagara (northern style), Vesara (mixed style), and Dravida (southern style), Indian vernacular architecture, Temple sytle, cave architecture, rock cut architecture, kalinga architecture, chandels architecture, rajput architecture, jain architecture, sikh architecture, Maratha architecture Indo-Islamic architectural, Indo-Saracenic revival architecture, Greco Buddhist style.

MODULE 4: Ancient Indian Science (Textile, Agriculture, Transport) — **9 hours**

Textile Technology in India: Cotton (natural cellulose fiber), silk, wool (natural protein fibers), bast and leaf fibers, mridhudhautadhupitambaram (meaning a practice of fumigating the fabric with incence smoke before use as a part of the finishing process), sitadhautavasanayugala (bleached white–a finishing process); suchhastah, sutradharah (needle and thread – tools for stiching). dyeing, washing spinning and weaving technology, Agriculture in India: krishisuktas,

Krishiparashara, Brihatsamhita, Types of crops, Manures, Types of land- devamatruka, nadimatruka, use of animals in warfare, animal husbandry, Animals for medicines. Ancient transport in India	
MODULE 5: Ancient Indian Science (Ayurveda & Yoga)	**7 hours**
Ayurveda for Life, Health and Well-being: Introduction to Ayurveda: understanding Human body and Pancha maha bhuta, the communication between body & mind, health regimen for wellbeing, introduction to yoga (raja yoga, astang yoga, gyan yoga), understanding of Indian psychological concept, consciousness, tridosha & triguna.	
Total Lecture hours	**42 hours**
Text Book/Reference Book:	
1. Textbook on IKS by Prof. B Mahadevan, IIM Bengaluru.	
2. Kapur K and Singh A.K (Eds) 2005). Indian Knowledge Systems, Vol. 1. Indian Institute of Advanced Study, Shimla. Tatvabodh of sankaracharya, Central chinmay mission trust, Bombay, 1995.	
3. Nair, Shantha N. Echoes of Ancient Indian Wisdom. New Delhi: Hindology Books, 2008.	
4. SK Das, The education system of Ancient hindus, Gyan publication house, India	
5. R P Kulkarni, Glimpese of Indian Engineering and Technology (Ancient & Medieval period, Munshiram Manoharlal Publishers Pvt. Ltd. 2018	
6. AK Pathak, Science and Technology in India, Anshika prakashan pratapgarh, 2016	
7. PB Sharma, S. Narain, Doctors Scientists and Engineers of Ancient India, Kalpaz Publications 2017	
8. NVP, Unithiri, Indian Scientific Traditions (Professor K.N. Neelakantan Elayath Felicitation Volume), publication division unieristy of Calicut, 2006	
9. Anonyms, History of Science in India- Volume-I Part-I (Physics, Mathematics and Statistics), the national academy of science, India & the ramkrishna mission institute of culture, 2014	
10. R N Basu, T K Bose, CS, Cakraborty History of Science in India - Agricultural Science (Volume V), the national academy of science, India & the ramkrishna mission institute of culture 2014	
11. A Gosh, History of Science in India (Volume-I Part-II Astronomy), the national academy of science, India & the ramkrishna mission institute of culture, 2014	
12. Dharmpal, Indian science and technology in the eighteen century, rashtrottahana sahitya, 1983	
13. S Biswal, B L ray, vedic Science and technology, DK Print world, 2009	
14. A.K Bag, Histroy of technology in Indian (Set 3 vol), Indian Nation Science Academy, 1997.	
15. AR vasudev Murty, Science and Technology in Ancient India as Reflected in the Mahabharata, Sanskrit bharati, 2019	