



Course: BTech

Semester: 4

Prerequisite: Basic understanding of computer system

Rationale: This course provides detail of computer system's functional components, their characteristics, performance and interactions including system bus, different types of memory and input/output organization and CPU. This course also covers the architectural issues such as instruction set program and data types. On top that, the students are also introduced to the increasingly important area of parallel organization. This course also serves as a basic to develop hardware-related projects. And hence it is an important course for all students of computer engineering branch.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
3	0	0	0	3	20	20	-	60	-	100	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%), **T** - Teaching hours

Sr.	Topics	W	T
1	Introduction to Microprocessor 8085 Microprocessor, Instruction set and computer languages, 8085 Programming Model, Instruction Data Format and storage.	10	4
2	Microprocessor architecture and interfacing Microprocessor architecture and its operations, Memory and I/O devices, Memory interfacing, Interfacing I/O devices	20	8
3	Programming methods with Instructions 8085 Instructions, Looping, Counting and Indexing, Logic operations Rotate and Compare	15	5
4	Additional Programming techniques Counter, time delay, Stack & Subroutines, Restart, Call and Return Instruction, Code conversion	15	8
5	8085 Interrupts Interrupt structure of 8085 microprocessor, processing of vectored and non-vectored interrupts, latency time and response time; Handling multiple interrupts	10	4
6	Computer Organization - Register Transfer and Basic Computer Design Register Transfer: Register Transfer language, Bus design using multiplexer and Tri-state buffer, Memory Transfers, Arithmetic Micro- Operations, Logic Micro-Operations, Shift Micro-Operations, Arithmetic logical shift unit. Basic Computer Design: Instruction codes, Computer registers, computer instructions, Timing and Control, Instruction cycle, Memory-Reference Instructions, Register Reference Instructions, IO Reference Instructions, Interrupt, Design of Accumulator Unit.	15	8
7	Computer Organization - Assembler and Memory Organization Assembler: Machine Language, Assembly Language, Assembler, Program loops, Programming Arithmetic and logic operations, subroutines, I-O Programming. Memory Organization: Memory hierarchy, Main memory, Auxiliary memory, Flash memory, Associative memory, Cache memory, Virtual memory	15	8



Reference Books

1.	Microprocessor Architecture, Programming, and Applications with the 8085 (TextBook) By Ramesh S. Gaonkar Penram International.
2.	Computer System Architecture By M.Morris Mano PHI 3rd Edition
3.	Microprocessor 8085 and its Interfacing By Sunil Mathur PHI Learning Pvt. Ltd
4.	8085 Microprocessor And its Applications By A. NagoorKani TMH Education Pvt. Ltd

Course Outcome

After Learning the Course the students shall be able to:

After learning this course students will be able to:

1. Explain 808microprocessor architecture.
2. Design assembly language program for 808microprocessor.
3. Design Interfacing with various hardware with 808microprocessor.
4. Debug program written in assembly language.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc



Course: BTech

Semester: 4

Prerequisite: Basic understanding of computer system

Rationale: This course provides detail of computer system's functional components, their characteristics, performance and interactions including system bus, different types of memory and input/output organization and CPU. This course also covers the architectural issues such as instruction set program and data types. On top that, the students are also introduced to the increasingly important area of parallel organization. This course also serves as a basic to develop hardware related projects. And hence it is an important course for all students of computer engineering branch.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
0	0	2	0	1	-	-	20	-	30	50	

SEE - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Outcome

After Learning the Course the students shall be able to:

After learning this course students will be able to:

1. Explain 808microprocessor architecture.
2. Design assembly language program for 808microprocessor.
3. Design Interfacing with various hardware with 808microprocessor.
4. Debug program written in assembly language.



List of Practical

1.	Addition of two 8 bit numbers using 8085 Addition of two 8 bit numbers using 8085
2.	Write a program to add two 16 bit numbers. Write a program to add two 16 bit numbers.
3.	Part A: write an 8085 assembly language to perform multiplication of two 8 bit nos. Part B: write an 8085 assembly language to perform division of two 8 bit nos. Part A: write an 8085 assembly language to perform multiplication of two 8 bit nos. Part B: write an 8085 assembly language to perform division of two 8 bit nos.
4.	Write a program to add blocks of 8-bit data stored in memory locations. Write a program to add blocks of 8-bit data stored in memory locations.
5.	Write an 8085 assembly language program to find the minimum from two 8-bit numbers Write an 8085 assembly language program to find the minimum from two 8-bit numbers
6.	Part A. Write an assembly language program to sort data in ascending order. Part B Write an assembly language program to sort data in descending order. Part A. Write an assembly language program to sort data in ascending order. Part B Write an assembly language program to sort data in descending order.
7.	Write an 8085 assembly language program to get the minimum from block of n 8-bit number Write an 8085 assembly language program to get the minimum from block of n 8-bit number
8.	Write a program to convert BCD to binary. Write a program to convert BCD to binary.
9.	Write a program to convert binary to BCD Write a program to convert binary to BCD
10.	Write an 8085 assembly language program to convert a given binary number into its equivalent ASCII number Write an 8085 assembly language program to convert a given binary number into its equivalent ASCII number
11.	Write an 8085 assembly language program to convert a given ASCII number into its equivalent binary number Write an 8085 assembly language program to convert a given ASCII number into its equivalent binary number

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.



Course: BTech

Semester: 4

Prerequisite: Fundamentals of Computer Systems

Rationale: This course is an introduction to the theory and practice behind modern computer operating systems. Topics will include what an operating system does (and doesn't) do, system calls and interfaces, processes, concurrent programming, resource scheduling and management, virtual memory, deadlocks, algorithms, programming, and security. The approach of the subject is from both a theoretical perspective as well as a practical one.

Teaching and Examination Scheme				
---------------------------------	--	--	--	--

Teaching Scheme					Examination Scheme				Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks		External Marks		Total	
					T	CE	P	T		
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content		W - Weightage (%) , T - Teaching hours	
Sr.	Topics	W	T
1	INTRODUCTION: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS-Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine.	5	3
2	PROCESSES, THREAD & PROCESS SCHEDULING: 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 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, FCFS, SJF, RR.	20	9
3	INTER-PROCESS COMMUNICATION: CriticalSection, 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, Dining Philosopher Problem etc	15	6
4	DEADLOCKS: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.	10	5
5	MEMORY MANAGEMENT & VIRTUAL MEMORY: 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).	30	13
6	I/O SYSTEMS, FILE & DISK MANAGEMENT: 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: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance. Disk Management: Disk structure, Disk scheduling algorithms - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks	20	9



Reference Books

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

Course Outcome

After Learning the Course the students shall be able to:

- After Learning the Course the students shall be able to:
1. Distinguish different styles of operating system design.
 2. Understand device and I/O management functions in operating systems as part of a uniform device abstraction.
 3. Understand disk organization and file system structure
 4. Give the rationale for virtual memory abstractions in operating systems.
 5. Understand the main principles and techniques used to implement processes and threads as well as the different algorithms for process scheduling.
 6. Understand the main mechanisms used for inter-process communication.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc



Course: BTech

Semester: 4

Prerequisite: Data Structures and Algorithms, Good working knowledge of C, and Fundamentals of Computer Systems.

Rationale: This course is an introduction to the theory and practice behind modern computer operating systems. Topics will include what an operating system does (and doesn't) do, system calls and interfaces, processes, concurrent programming, resource scheduling and management, virtual memory, deadlocks, and algorithms, programming, and security. We will approach the subject from both a theoretical perspective as well as a practical one

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
0	0	2	0	1	-	-	20	-	30	50	

SEE - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Outcome

After Learning the Course the students shall be able to:

1. Experiment with Linux commands and shell programming.
2. Able to build shell program for process and file system management with system calls.
3. Able to implement and analyse the performance of CPU scheduling algorithm.
4. Able to implement and analyse the performance of page replacement algorithms.
5. Able to implement and analyse the performance of deadlock avoidance and detection algorithm.

List of Practical

1.	Study of Basic commands of Linux.
2.	Study the basics of shell programming.
3.	Write a Shell script to print given numbers sum of all digits.
4.	Write a shell script to validate the entered date. (eg. Date format is: dd-mm-yyyy).
5.	Write a shell script to check entered string is palindrome or not.
6.	Write a Shell script to say Good morning/Afternoon/Evening as you log in to system.
7.	Write a C program to create a child process.
8.	Finding out biggest number from given three numbers supplied as command line arguments.
9.	Printing the patterns using for loop.
10.	Shell script to determine whether given file exist or not.
11.	Write a program for process creation using C. (Use of gcc compiler).
12.	Implementation of FCFS &Round Robin Algorithm.
13.	Implementation of Banker's Algorithm.



Course: BTech

Semester: 4

Prerequisite: Basic knowledge of software applications

Rationale: This course provides a broad introduction to software engineering. The various process models required to develop software is also being described. Moreover the functional and non-functional requirements are also described

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
3	0	0	0	3	20	20	-	60	-	100	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	Introduction: Study of Different Models, Software Characteristics, Components, Applications, Layered Technologies, Processes, Methods and Tools, Generic View Of Software Engineering, Process Models- Waterfall model, Incremental, Evolutionary process models- Prototype, Spiral And Concurrent Development Model Agile Development : Agility and Agile Process model, Extreme Programming, Other process models of Agile Development and Tools.	10	6
2	Software Project Management: Management Spectrum, People 'Product 'Process- Project, W5HH Principle, Importance of Team Management Planning a Software Project : Scope and Feasibility, Effort Estimation, Schedule and staffing, Quality Planning, Risk management- identification, assessment, control, project monitoring plan, Detailed Scheduling	10	5
3	Requirements Engineering: Problem Recognition, Requirement Engineering tasks, Processes, Requirements Specification, Use cases and Functional specification, Requirements validation, Requirements Analysis	10	5
4	Structured System Design: Design Concepts, Design Model, Software Architecture, Data Design, Architectural Styles and Patterns, Architectural Design, Alternative architectural designs, Modeling Component level design and its modeling, Procedural Design, Object Oriented Design. Data Oriented Analysis & Design : Difference between Data and Information, E-R Diagram, Dataflow Model, Control Flow Model, Control and Process Specification, Data Dictionary	15	5
5	Coding and Unit Testing: Programming principles and guidelines, Programming practices, Coding standards, Incremental development of code, Management of code evaluation, Unit testing- procedural units, classes, Code Inspection, Metrics- size measure, complexity metrics, Cyclomatic Complexity, Halstead measure,Knot Count, Comparison Of Different Metrics	10	4
6	Software Testing and Quality Assurance: Concepts, Psychology of testing, Levels of testing, Testing Process- test plan, test case design, Execution, Black-Box testing 'Boundary value analysis 'Pair wise testing- state based testing, White-Box testing criteria and test case generation and tool support Quality Assurance : Quality Control, Assurance, Cost, Reviews, Software Quality Assurance, Approaches to SQA, Reliability, Quality Standards- ISO9000 And 9001	15	7
7	CASE Tools and Advance Practices of System Dependability and Security: Computer Aided Software Engineering Tools, SCRUM Developments, Dependable System, Reliability Engineering, Safety Engineering, Security Engineering, Resilience Engineeirng	15	5



8	Advance Software Engineering: Software Reuse, Component Based Software Engineering, Distributed Software Engineering, Service-Oriented Software Engineering, Real-Time Software Engineering, Systems Engineering, Systems of System.	15	5
----------	--	-----------	----------

Reference Books

1.	Software Engineering (TextBook) R.Pressmen; 6th (TextBook)
2.	Software Engineering By Sommerville
3.	Fundamentals of Software Engineering By Rajib Mall PHI
4.	Software Engineering By Pankaj Jalote Wiley India

Course Outcome

After Learning the Course the students shall be able to:

After learning this course students will be able to :

1. Prepare and perform Software Requirement Specification and Software Project Management Plan.
2. Ensure the quality of software product, different quality standards and software review techniques
3. Apply the concept of Functional Oriented and Object Oriented Approach for Software Design.
4. Understand modern Agile Development and Service Oriented Architecture Concept of Industry
5. Analyze, design, verify, validate, implement and maintain software systems.
6. Execute a Project Management Plan, tabulate Testing Plans and Reproduce effective procedures.



Course: BTech

Semester: 4

Prerequisite: Basic knowledge of software applications.

Rationale: This course provides a broad introduction to software engineering. The various process models required to develop software is also being described. Moreover the functional and non-functional requirements are also described.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
0	0	2	0	1	-	-	20	-	30	50	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Outcome

After Learning the Course the students shall be able to:

After learning this course students will be able to :

1. Prepare and perform Software Requirement Specification and Software Project Management Plan.
2. Ensure the quality of software product, different quality standards and software review techniques
3. Apply the concept of Functional Oriented and Object Oriented Approach for Software Design.
4. Understand modern Agile Development and Service Oriented Architecture Concept of Industry
5. Analyze, design, verify, validate, implement and maintain software systems.
6. Execute a Project Management Plan, tabulate Testing Plans and Reproduce effective procedures.

List of Practical

1.	Project Definition and objective of the specified module and Perform Requirement Engineering Process.
2.	Identify Suitable Design and Implementation model from the different software engineering models.
3.	Prepare Software Requirement Specification (SRS) for the selected module.
4.	Develop Software project management planning (SPMP) for the specified module.
5.	Do Cost and Effort Estimation using different Software Cost Estimation models.
6.	Prepare System Analysis and System Design of identified Requirement specification using structure design as DFD with data dictionary and Structure chart for the specific module.
7.	Designing the module using Object Oriented approach including Use case Diagram with scenarios, Class Diagram and State Diagram, Collaboration Diagram, Sequence Diagram and Activity Diagram.
8.	Defining Coding Standards and walk through.
9.	Write the test cases for the identified module.
10.	Demonstrate the use of different Testing Tools with comparison.
11.	Define security and quality aspects of the identified module.



Course: BTech

Semester: 4

Prerequisite: knowledge of Computer and Information system

Rationale: This course is design to provide the basic knowledge about the data & signals. It also provides basic concepts of computer network and firm foundation for understanding how data communication occurs in the Transmission Medium. It will help to develop logical abilities and practically setup the network .

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
3	0	0	0	3	20	20	-	60	-	100	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	DATA COMMUNICATION COMPONENTS: Representation of data and its flow Networks, Various Connection Topology, Protocols and Standards, OSI model, Transmission Media, LAN: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN, Techniques for Bandwidth utilization: Multiplexing - Frequency division, Time division and Wave division, Concepts on spread spectrum	25	11
2	DATA LINK LAYER AND MEDIUM ACCESS SUB LAYER: Error Detection and Error Correction - Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, Goback 'N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols - Pure ALOHA, Slotted ALOHA, CSMA/CD, CDMA/CA	25	11
3	Network Layer: Switching, Logical addressing 'IPV4, IPV6; Address mapping 'ARP, RARP, BOOTP and DHCP' Delivery, Forwarding and Unicast Routing protocols	20	8
4	Transport Layer: Process to Process Communication, User Datagram Protocol(UDP), Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm.	15	6
5	Application Layer: Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography	15	6

Reference Books

1.	Computer Networks (TextBook) By Andrew S. Tanenbaum and David J. Wetherall PEARSON Edition
2.	Internetworking with TCP/IP Principles, Protocols and Architecture By Douglas E Comer
3.	TCP/IP Illustrated By Richard Stevens
4.	Data Communication and Networking By Behrouz A. Forouzan
5.	"Data and computer communications", By William Stallings Prentice Hall



Course Outcome

After Learning the Course the students shall be able to:

After Learning the course the students shall be able to:

1. Draw the functional block diagram of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANS) describe the function of each block.
2. Understand the functions of the different layers of the OSI Protocol
3. Understand and Design For a given requirement (small scale) of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANS) design it based on the market available component
4. Learn on the given problem-related TCP/IP protocol developed for the network programming.
5. Configure DNS DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, and Firewalls using open-source available software and tools.



Course: BTech

Semester: 4

Prerequisite: knowledge of Computer and Information system

Rationale: This course is designed to provide basic knowledge about the data & signals. It also provides basic concepts of computer networks and a firm foundation for understanding how data communication occurs in the Transmission Medium. It will help to develop logical abilities and practically set up the network.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
0	0	2	0	1	-	-	20	-	30	50	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Outcome

After Learning the Course the students shall be able to:

After Learning the Course the students shall be able to:

- Configure and set up different types of networks, including local area networks (LANs) and wide area networks (WANs).
- Configure routers and switches, and implement routing protocols to understand how data is directed through a network.
- Use network monitoring tools to analyze network.
- Apply security measures, such as firewalls, encryption, and intrusion detection systems, to secure network communication.
- Implement and analyze various network protocols, such as TCP/IP, UDP, and ICMP, through practical exercises.

List of Practical

1.	Experiments on Simulation Tools: (CISCO PACKET TRACER).
2.	Experiments of Packet capture tool: Wireshark.
3.	To study behavior of generic devices used for networking: (CISCO PACKET TRACER).
4.	Data Link Layer (Error Correction).
5.	Virtual LAN
6.	Wireless LAN
7.	Inter networking with routers: 1: Experiment on same subnet 2: Perform Experiment across the subnet and observe functioning of Router via selecting suitable pair of Source and destination.
8.	Implementation of SUBNETTING.
9.	Routing at Network Layer.
10.	Experiment on Transport Layer.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.



Course: BTech

Semester: 4

Prerequisite: Basic knowledge of Programming and web applications

Rationale: This course provides a broad introduction to Python programming and development of web applications. Developing and using Python as a scripting language for automating tasks and data processing. Moreover Building and deploying web applications using popular Python frameworks such as Django and Flask.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
3	0	0	0	3	20	20	-	60	-	100	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	Introduction to python programing: Introduction to Python and basic programming concepts, variables, data types, conditionals statements and loops Lists,Sets,Tuples,Dictionaries: Working with strings, lists, sets, tuples and dictionaries, including common operations and built-in functions	15	6
2	Functions : Defining and using functions, including the use of arguments and return values OOPS Concepts : Object, class, abstraction, encapsulation, polymorphism, Inheritance. Exceptions and File handling: Handling exceptions and working with files	20	5
3	Modules and Packages: Working with modules and packages in Python Introduction to popular Python libraries for specific tasks, such as data analysis, web development, or game development. PyCharm IDE : GIT- Git Integration with PyCharm IDE, PyTests. Python connectivity with Databases MYSQL, MongoDB CRUD operations.	15	5
4	Flask Framework: Introduction to Flask and web development with Python, Installation in Virtual Environment. Creation Routing App Settings URL Building HTTP methods Templates Working with Static, Media Files. Sending Form Data to Template. Flask App with Database connectivity SQLite3, MySQL. Handling Exceptions and Errors Flash Message Working with Mails. Authenticating and authorizing users with Flask-Login, Deploying a Flask application to a web server.	20	10
5	Django Framework: Introduction to Django framework, Django Project Installation in Virtual Environment. Phases in Django Project Creation Create a Project. Creation of Apps and their Structure. Working with ADMIN Console. Creating Views URL Mapping. Template System Working with Models. Form Processing static, media files, Django App Deployment.	20	10
6	RESTful APIs: Introduction to RESTful APIs and the REST architectural style Understanding the HTTP protocol and its role in RESTful APIs Designing and implementing RESTful APIs using common HTTP methods, such as GET, POST, PUT, and DELETE Using URLs and resource representations to identify and transfer data in RESTful APIs Implementing best practices for designing and implementing RESTful APIs, such as using HTTP status codes,	10	6



versioning, and error handling		
Consuming RESTful APIs using common tools and libraries, such as cURL, Postman, and the requests library in Python		
Building scalable and secure RESTful APIs using common frameworks and libraries Flask or FastAPI.		

Reference Books

1. **Fluent Python, 2nd Edition by Luciano Ramalho (TextBook)**
2. **Learn Python3 the Hard Way By Zed Shaw**
3. **"Django for Beginners: Build websites with Python and Django" by William S. Vincent.**
4. **"Learning Django Web Development" by Samuli Natri.**
5. **"Flask Web Development with Python" by Miguel Grinberg.**
6. **"Mastering Flask" by Jack Stouffer.**
7. **"Building RESTful Python Web Services" by Gastón C. Hillar.**
8. **"Building Web APIs with FastAPI" by Samuel Colvin.**

Course Outcome

After Learning the Course the students shall be able to:

After learning this course students are able to:

1. Understand the fundamental concepts of web development.
2. Create and manipulate data using a variety of databases, including SQL and NoSQL
3. Build and deploy web applications using a popular Python web framework, such as Django or Flask.
4. Design and implement APIs (application programming interfaces) that enable different applications to communicate with each other.
5. Test and debug web applications, and to deploy them to production environments.



Course: BTech

Semester: 4

Prerequisite: Basic knowledge of Programming and web applications

Rationale: This course provides a broad introduction to Python programming and development of web applications. Developing and using Python as a scripting language for automating tasks and data processing. Moreover Building and deploying web applications using popular Python frameworks such as Django and Flask

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
0	0	2	0	1	-	-	20	-	30	50	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Outcome

After Learning the Course the students shall be able to:

After Learning the Course the students shall be able to:

1. Demonstrate a strong understanding of Python programming language fundamentals, including syntax, data types, control structures, and functions.
2. Understand the basics of web development, including HTML, CSS, and JavaScript, and demonstrate the ability to create static web pages.
3. Design and implement RESTful APIs using Python for communication between the front-end and back-end components.
4. Identify and resolve issues in both front-end and back-end code, and optimize the performance of web applications.
5. Integrate AJAX techniques into Django applications to enable dynamic updates and improve interactivity without full page reloads.



List of Practical

1.	Set-1 <ol style="list-style-type: none">1. A program that converts temperatures from Fahrenheit to Celsius and vice versa.2. A program that calculates the area and perimeter of a rectangle.3. A program that generates a random password of a specified length.4. A program that calculates the average of a list of numbers.5. A program that checks if a given year is a leap year.6. A program that calculates the factorial of a number.7. A program that checks if a given string is a palindrome.8. A program that sorts a list of numbers in ascending or descending order.9. A program that generates a multiplication table for a given number.10. A program that converts a given number from one base to another.
2.	Set-2 <ol style="list-style-type: none">1. A program that models a bank account, with classes for the account, the customer, and the bank.2. A program that simulates a school management system, with classes for the students, the teachers, and the courses.3. A program that reads a text file and counts the number of words in it.4. A program that reads a CSV file and calculates the average of the values in a specified column.5. A program that reads an Excel file and prints the data in a tabular format.
3.	Set-3 <ol style="list-style-type: none">1. A program that creates a simple web server and serves a static HTML page.2. A program that creates a web application that allows users to register and login.3. A program that creates a web application that allows users to upload and download files.4. A program that creates a web application that displays data from a database in a tabular format.5. A program that creates a web application that accepts user input and sends it to a server-side script for processing.
4.	Set-4 <ol style="list-style-type: none">1. A program that creates a web application that uses a template engine to generate dynamicHTML pages.2. A program that creates a web application that supports AJAX requests and updates the page without reloading.3. A program that creates a web application that uses Django's built-in debugging features to troubleshoot errors and exceptions.4. A program that creates a web application that implements user authentication and authorization.5. A program that creates a web application that integrates with third-party APIs to provide additional functionality.
5.	Set-5 <ol style="list-style-type: none">1. A program that creates a simple RESTful API that returns a list of users in JSON format.2. A program that creates a RESTful API that allows users to create, read, update, and delete resources.3. A program that creates a RESTful API that authenticates users using a JSON Web Token.4. A program that creates a RESTful API that paginates the results of a query to improve performance.5. A program that creates a RESTful API that supports data validation and error handling.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.



Course: BTech

Semester: 4

Prerequisite: proficiency in a programming language (e.g., C++, Python) and a strong grasp of data structures and algorithms, with a focus on problem-solving skills and efficient code implementation. Familiarity with common coding platforms (e.g., Codeforces, LeetCode) is also beneficial.

Rationale: Competitive coding sharpens problem-solving skills, enhances algorithmic thinking, and fosters quick and efficient coding practices. It provides a platform for continuous learning, challenges individuals to tackle diverse problems, and fosters a competitive spirit that's valuable in technical interviews and real-world software development.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
-	-	4	-	2	-	-	20	-	30	50	

SEE - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Outcome

After Learning the Course the students shall be able to:

After Learning the Course the students shall be able to:

1. Develop strong problem-solving skills, improve algorithmic thinking, and enhance proficiency in coding by tackling a variety of challenging problems.
2. Cultivate the ability to write efficient and optimized code under time constraints, honing the skill of quickly translating algorithmic insights into practical solutions.
3. Gain a competitive advantage in technical interviews and coding assessments, showcasing the ability to tackle diverse coding challenges commonly encountered in job placements and coding competitions.
4. Foster a mindset of continuous learning by regularly engaging with new problems, staying updated on emerging algorithms, and adapting to evolving coding paradigms.

List of Practical

1.	Write a program for implementing a MINSTACK which should support operations like push, pop, overflow, underflow, display 1. Construct a stack of N-capacity 2. Push elements 3. Pop elements 4. Top element 5. Retrieve the min element from the stack
2.	Write a program to deal with real-world situations where Stack data structure is widely used Evaluation of expression: Stacks are used to evaluate expressions, especially in languages that use postfix or prefix notation. Operators and operands are pushed onto the stack, and operations are performed based on the LIFO principle.
3.	Write a program for finding NGE NEXT GREATER ELEMENT from an array.
4.	Write a program to design a circular queue(k) which Should implement the below functions a. Enqueue b. Dequeue



	c. Front d. Rear
5.	Write a Program for an infix expression, and convert it to postfix notation. Use a queue to implement the Shunting Yard Algorithm for expression conversion.
6.	Write a Program for finding the Product of the three largest Distinct Elements. Use a Priority Queue to efficiently find and remove the largest elements.
7.	Write a Program to Merge two linked lists(sorted).
8.	Write a Program to find the Merge point of two linked lists(sorted).
9.	Write a Program to Swap Nodes pairwise.
10.	Write a Program for Building a Function ISVALID to VALIDATE BST.
11.	Write a Program to Build BST.
12.	Write a Program to determine the depth of a given Tree by Implementing MAXDEPTH.
13.	Write a Program to Understand and implement Tree traversals i.e. Pre-Order Post-Order, In-Order.
14.	Write a Program to perform Boundary Traversal on BST.
15.	Write a program for Lowest Common Ancestors.
16.	Write a Program to verify and validate mirrored trees or not.
17.	Write a Program for a basic hash function in a programming language of your choice. Demonstrate its usage to store and retrieve key-value pairs.
18.	Implement a hash table using separate chaining for collision handling. Perform operations like insertion, deletion, and search on the hash table.
19.	



	Write a Program to Implement Two sums using HASHMAP.
20.	Write a Program to Implement Search, insert, and Remove in Trie.
21.	Write a Program to Implement Huffman coding.
22.	Write a Program to find Distinct substrings in a string.
23.	Write a Program to find The No of Words in a Trie.
24.	Write a Program to view a tree from left View.
25.	Write a Program to Traverse a Tree using Level Order Traversal.



Course: BTech

Semester: 4

Prerequisite: Fundamentals of networking.

Rationale: The objective of this subject is to train the students about various types of pen testing methodology for network, basic concepts of penetration testing of wired and wireless networks.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
3	0	0	-	3	20	20	-	60	-	100	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%) , T - Teaching hours

Sr.	Topics	W	T
1	Types of Network & OSI Layers: Types of networks, IP Address, NAT, IP Subnets, DHCP Server, Ports, DNS, Proxy Servers, DNS Server, OSI and TCP IP Model	15	6
2	Basics of Routing & Switching: Routers, Switches, Endpoint solutions, Access Directory, TOR Network. Networking Devices (Layer1,2,3) - Different types of network layer attacks–Firewall, ACL, Packet Filtering, DMZ, Alerts and Audit Trails	20	9
3	Proxy & VPNs: PN and its types –Tunnelling Protocols – Tunnel and Transport Mode –Authentication Header- IPSEC Protocol Suite – IKE PHASE 1, Implementation of VPNs.	20	10
4	Wireless Attacks: Network Sniffing, Wireshark, packet analysis, display and capture filters, ettercap, DNS Poisoning, Denial of services, Vulnerability scanning, Nessus, Network Policies, Network Scanning Report Generation, Router attacks, Packet Sniffing, Types of authentication, Fake Authentication Attack, De authentication, Attacks on WPA and WPA-2 Encryption, fake hotspot, WPA & WPA-2 attacks, Wireless Hacking using phishing, MITM (man in the middle attack), Brute Force Attacks	25	10
5	Network pentest: HOST DISCOVERY, PORT SCANNING., Banner Grabbing/OS Fingerprinting., Scan for Vulnerabilities., Draw Network Diagrams., Prepare Proxies., Document all Findings.	20	10

Reference Books

1.	Network Security, Private communication in public world (2nd Ed.) PHI - Kaufman, C., Perlman, R., & Speciner, M. (TextBook)
2.	Network Security. Wiley - Perez, Andre.
3.	Cryptography and Network Security: Principles and Practice (5th Ed.). Prentice Hall - Stallings, W.
4.	Network Attacks and Exploitation: A Framework. Wiley - Monte, M.
5.	Network Security Essentials William Stallings
6.	Computer Networking: A Top-Down Approach James F. Kurose, Keith W. Ross



Course Outcome

After Learning the Course the students shall be able to:

After Learning the Course the students shall be able to:

1. Learn networking concepts and enhance their knowledge of different network devices.
2. Learn and apply different types of network and wireless attacks and their countermeasures.
3. Executing vulnerability assessment and penetration testing on IT Landscape of organization.
4. Prepare VAPT reports.
5. Enhance their skills on Patch Management of IT Organization.



Course: BTech

Semester: 4

Prerequisite: Fundamentals of networking

Rationale: The objective of this subject is to train the students about various types of pen testing methodology for network, basic concepts of penetration testing of wired and wireless networks.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
0	0	2	-	1	-	-	20	-	30	50	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Outcome

After Learning the Course the students shall be able to:

After Learning the Course the students shall be able to:

1. Learn networking concepts and enhance their knowledge of different network devices.
2. Learn and apply different types of network and wireless attacks and their countermeasures.
3. Executing vulnerability assessment and penetration testing on IT Landscape of organization.
4. Prepare VAPT reports.
5. Enhance their skills on Patch Management of IT Organization.

List of Practical

1.	Introduction of cisco packet tracer.
2.	Create a logical network diagram with eight PCs and switch in cisco packet tracer which are in same network and check for the communication.
3.	1. Create a logical network diagram with two different networks, each network contains two pc, one switch and one router. 2. Configure the routing on that scenario. 3. check the connectivity between different network devices.
4.	Perform Man in Middle Attack for DNS spoofing and ARP using Ettercap tool.
5.	Setup a VPN in windows operating system.
6.	Setup a Proxy in windows operating system.
7.	Perform the Wireless recon.
8.	Perform the network vulnerability scanning using Nessus tool
9.	Perform the NTLM based Brute Force Attack.
10.	Perform the network sniffing using Wireshark.
11.	Perform the basic network scanning using Nmap tool.
12.	Finding the live host in network using advance IP scanning tool.



Course: BTech

Semester: 4

Prerequisite: Fundamentals of Cryptography

Rationale: The objective of this subject is to train students in the fundamentals of cryptography, including both symmetric and asymmetric encryption methods. The course will focus on secure communication techniques, cryptography algorithms, digital signatures, and key management, providing a solid foundation for securing data in networked environments.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
3	0	0	0	3	20	20	-	60	-	100	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	Introduction to Cryptography and Traditional Ciphers Security Goals, Attacks, Services, Mechanisms, Techniques, Traditional Symmetric-Key Ciphers: Substitution Cipher, Transposition Cipher, Stream and Block Ciphers.	20	6
2	Modern Symmetric-Key Ciphers and DES Introduction to Modern Block and Stream Ciphers, Data Encryption Standard (DES), DES Structure, Analysis, Multiple DES, Advanced Encryption Standard (AES), AES Structure and Analysis.	20	12
3	Asymmetric-Key Cryptography & Message Authentication Introduction to RSA, RABIN, ELGAMAL Cryptosystems, Message Integrity and Authentication, Cryptographic Hash Functions (SHA-512, MD5)	20	12
4	Digital Signatures & Key Management RSA and ELGAMAL Digital Signature Schemes, Entity Authentication: Passwords, Challenge–Response, Zero–Knowledge Protocols, Symmetric-Key Distribution, KERBEROS	20	10
5	Security at Application and Transport Layer PGP, S/MIME, SSL Architecture	20	5

Reference Books

1.	Cryptography and Network Security Principles and Practices (TextBook) By Williams Stallings Pearson Education Third Edition
2.	Cryptography & Network Security, Atul Kahate, The McGraw-Hill Companies.
3.	The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations by Gene Kim, Jez Humble, Patrick Debois, and John Willis

Course Outcome

After Learning the Course the students shall be able to:

- Identify the fundamentals of cryptography, including symmetric and asymmetric key algorithms.
- Analyze and implement traditional and modern cryptography algorithms such as DES, AES, and RSA.
- Apply cryptography techniques to ensure message integrity and authentication.
- Discuss the role of cryptography in secure communication at the application and transport layers.
- Develop skills in managing keys, securing digital communications, and working with protocols like PGP, S/MIME, and SSL.



Course: BTech

Semester: 4

Prerequisite: Basic knowledge of Fundamentals of Networking, Computer Programming.

Rationale: The objective of this lab offers hands-on experience with key cryptographic algorithms, focusing on encryption, key exchange, message integrity, and secure communication. It equips students with the skills to apply cryptography in real-world cybersecurity scenarios.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
0	0	2	-	1	-	-	20	-	30	50	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Outcome

After Learning the Course the students shall be able to:

1. Identify and implement classical cryptographic techniques and analyze their vulnerabilities.
2. Apply modern cryptographic algorithms for secure key exchange and encryption.
3. Implement public-key cryptosystems and understand their real-world applications.
4. Describe and implement cryptographic hashing and ensure message integrity.
5. Apply cryptographic techniques to ensure secure communication, key management, and signatures.



List of Practical

1.	Practical-1 Implement ceaser cipher and apply brute force attack to get original key.
2.	Practical-2 Apply attacks for cryptanalysis to decrypt the original message from a given cipher text using Play fair cipher. Key = Parul
3.	Practical-3 Implement Deffi Hellmn key exchange algorithm. Generate share secret without Sharing the secret code.
4.	Practical-4 Implement and Analyze DES algorithm.
5.	Practical-5 Implement RSA cryptosystem.
6.	Practical-6 Implement Message Integrity using SHA-256 hashing function which creates chain of three blocks. Each block contains index, timestamp, data, previous hash value and current block of hash value. Test message integrity of program by modifying one of the hash value of block.
7.	Practical-7 Implement Elgamal Cryptosystem.
8.	Practical-8 Implement Digital Signature.
9.	Practical-9 Study and Configure SSH to authenticate machines, generate session key and transfer files using symmetric key and asymmetric key cryptography.
10.	Practical-10 Prepare report on any advanced cryptographic system.



Course: BTech

Semester: 4

Prerequisite: Basic concepts of Statistics and Probability.

Rationale: The course provides systematic knowledge of probability, numerical and statistical methods.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
4	-	-	-	4	20	20	-	60	-	100	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	UNIT 1Correlation, Regression and Curve fitting: Correlation and Regression – Rank correlation Curve Fitting by The Method of Least Squares- Fitting of Straight Lines, Second Degree Parabolas and More General Curves.	18	11
2	UNIT 2Probability and Probability Distributions: Probability Spaces, Conditional Probability, Bayes' Rule, Discrete and continuous random variables, Independent Random Variables, Expectation and Variance of Discrete and Continuous Random Variables, Distribution and Their Properties: Binomial Distribution, Poisson Distribution, Normal Distribution.	23	13
3	UNIT 3Testing of Hypothesis: Test of significance: Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations. Test for single mean, difference of means, Test for ratio of variances, Chi-square test for goodness of fit and independence of attributes.	26	15
4	UNIT 4Solution of a System of Linear Equations, Roots of Algebraic and Transcendental Equations: Gauss-Jacobi and Gauss Seidel Methods, Solution of Polynomial and Transcendental Equations – Bisection Method, Newton-Raphson Method and Regula-Falsi Method.	11	7
5	UNIT 5Finite Differences and Interpolation: Finite Differences, Relation between Operators, Interpolation using Newton's Forward and Backward Difference Formulae. Newton's Divided and Lagrange's Formulae for Unequal Intervals.	11	7
6	UNIT 6Numerical Integration: Trapezoidal rule, Simpson's 1/3rd and 3/8th Rules, Gaussian Quadrature Formulae. Numerical solution of Ordinary Differential Equations: Taylor's Series, Euler and Modified Euler's Methods. Runge-Kutta Method of Fourth Order for Solving First and Second Order Equations.	11	7

Reference Books

1.	Introductory Methods of Numerical Analysis By Sastry S. S Prentice Hall of India
2.	Numerical Methods in Engineering & Science with Programs in C and C++ (TextBook) By Dr. B. S. Grewal Khanna Publishers
3.	Introduction to Numerical Analysis By C.E. Froberg Addison Wesley Publishing Company
4.	Introduction to Probability (TextBook) By P. G. Hoel, S. C. Port and C. J. Stone, UBS Publishers,
5.	Fundamentals of Mathematical Statistics (TextBook) By S.C. Gupta and V. K. Kapoor Sultan Chand & Sons



Course Outcome

After Learning the Course the students shall be able to:

- Analyse correlation and regression between two variables and fit a curve to the given set of values.
- Calculate probabilities and analyse random variables to determine expectation and variance.
- Evaluate hypotheses by conducting significance tests for proportions, means, standard deviations, and variances using large sample tests, chi-square tests, and other appropriate statistical methods.
- Apply numerical methods such as Gauss-Jacobi, Gauss Seidel, bisection method, Newton-Raphson method, and Regula-Falsi method to solve systems of linear equations and algebraic/transcendental equations
- Interpolate data using finite differences and various interpolation techniques including Newton's forward/backward difference formulae, and Lagrange's formulae for unequal intervals.
- Utilize numerical integration techniques such as the trapezoidal rule, Simpson's rules, and Gaussian quadrature formulae, as well as numerical methods including Taylor's series, Euler's method, Modified



Course: BTech

Semester: 4

Prerequisite: Knowledge of communication theories and basic management skills are essential.

Rationale: Acquiring soft skills, life skills & aptitude skills are crucial for organizational communication as well as for employability respectively.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total	
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks			
					T	CE	P	T	P		
-	1	-	-	1	100	100	-	-	-	100	

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%) , T - Teaching hours

Sr.	Topics	W	T
1	Self Development and Assessment Various self-assessments for personal and professional development skills that are relevant to career development: - Change, Grow, Persist, Prioritize, Read, Learn, Listen, Record, Remember, Guess, Think, Communicate, Relate, and Dream	25	4
2	Corporate Etiquette Tips and guide to develop personality and gain various etiquettes manners, case studies and activities. Telephone etiquette Etiquette for foreign business trips Etiquette for small talks Respecting privacy Learning to say 'No'	25	4
3	Public Speaking It's process of communicating information to an audience and is helpful in career advancement. Effective Public speaking skills includes: Choosing appropriate pattern Selecting appropriate method Art of persuasion Making speeches effective Delivering different types of speeches	20	4
4	Reading Skills Activity & Reading Comprehension Aims to improve students' Comprehensive Skills in English Language by getting them involved in reading activity and providing practice for reading comprehension.	15	2
5	Listening Skills- Inquiry Based Listening Questions Aims to improve students' listening skills in English Language providing them practice of various types of inquiry based listening tracks. Students will listen and will be able to find out details from the conversations.	15	1

Course Outcome

After Learning the Course the students shall be able to:

After Learning the course the students shall be able to:

1. Identity and develop soft skills required for personal and professional growth.
2. Develop professional etiquette & desired behaviour at the workplace
3. Speak and participate effectively in oral organizational communication
4. Improve comprehensive skills for reading
5. Know how to be assertive in professional environment