SEMESTER V

Course Code: **CSE310C**	**Operating System**	Credits: **03** L – **3** P – **0**

Course Outcomes (COs):

- Describe the detailed functioning of an operating system.
- Describe the various types, models and structures of operating systems.
- Describe how processes and threads perform inter-process and inter-thread communication.
- Describe and simulate the policies for process synchronization, scheduling and deadlocks.
- Describe and simulate memory management, file management and disk management systems.

Unit – I (7)

Introduction: What is an operating system, history of operating systems, types of operating systems, operating system concepts, functions of an operating system, kernel data structures, computing environments, open-source operating systems, user and operating-system interface, system calls, types of system calls, operating-system design and implementation, system programs, operating system structures, operating-system debugging, system boot, programming projects.

Unit – II (10)

Processes: Process concept, Process scheduling, Operations on processes, Interprocess communication.
Threads: Overview, Multicore Programming, Multithreading Models.
CPU Scheduling: Scheduling criteria, Scheduling algorithms.

Unit – III (10)

Process Synchronization: Concurrency, The Critical-Section problem, Peterson's solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic problems of synchronization – The Bounded-Buffer problem, The Readers-Writers problem, The Dining-Philosophers problem, Monitors.
Deadlocks: Characterization, prevention, avoidance, detection, recovery.

Unit – IV (10)

Main Memory Management: Background, Swapping, Contiguous memory allocation, Segmentation, Paging, Structure of the page table.
Virtual Memory Management: Background, Demand Paging, Copy-on-write, Page Replacement, Allocation of Frames, Thrashing.

Unit – V (8)

Mass-Storage Management: Disk scheduling, Disk management, Swap-space management, RAID.
File System Interface: File concept, File access methods, Directory and Disk Structure, File Sharing, protection.
File-System Implementation: File-System Structure, File-System Implementation, Directory Implementation, File Allocation Methods, Free-Space Management, NFS.

Textbooks:

1. Silberschatz, Abraham, Peter Baer Galvin, and Greg Gagne. *Operating system principles*. 9th Edition, John Wiley & Sons.

Reference Books:

1. Tanenbaum, Andrew S., and Bos Herbert. *Modern operating systems. 4th Edition, Pearson.*
2. Stallings, William, and Goutam Kumar Paul. *Operating systems: internals and design principles*. 9th Edition, Pearson, 2012.
3. Tanenbaum, Andrew S., and Albert S. Woodhull. *Operating systems: design and implementation*. Prentice Hall, 1997.
4. Crowley, Charles. *Operating systems: a design-oriented approach*. McGraw-Hill Professional.
5. Nutt, Gary J. *Operating Systems: A Modern Perspective, Lab Update*. Addison-Wesley Longman Publishing Co., Inc

Online Resources:

1. https://os-book.com/OS10/index.html
2. https://nptel.ac.in/courses/106105214

Course Code: **CSE311C**	**Computer Networks**	Credits: **03** L – **3** P – **0**

Course Outcomes (COs):

- Understanding of the fundamental concepts of computer networks.
- Demonstrate proficiency in the data link layer, including knowledge of various design issues.
- Should have a deep comprehension of the network layer like subnetting and routing mechanisms.
- Able to explain and work with key internetworking protocols.
- Students should have a strong grasp of the application layer.

Unit – I **(6)**

Introduction: Uses of Computer Networks, Network and Protocol Architecture, Reference Model (ISO-OSI, TCP/IP-Overview), Types of networks (LAN, MAN and WAN), Different network topologies like star, ring, hybrid, tree etc. IEEE standards.

Unit – II **(10)**

Data Link layer – Design Issues, Error detection and Correction techniques, Flow control algorithms, Framing techniques, Switched LANs: L2 addressing and ARP, Ethernet frame structure, learning switches. Medium access protocols: Polling vs. contention-based: TDM, Aloha, CSMA/CD.

Unit – III **(10)**

Network Layer: Network architecture and Performance, Need for an Internet address, and its design: IPv4 and IPv6, structure of IP datagram, IP forwarding, NATs, sub-netting, Routing protocols: Link state routing. Distance vector routing: count-to-infinity, routing convergence.

Unit – IV **(10)**

Internetworking: Internet control protocols: ICMP, ARP, RARP, BOOTP, DHCP, intra-domain (OSPF) and inter-domain (BGP)routing.
Transport Layer: Transport-Layer Services, Multiplexing and Demultiplexing, Connectionless Transport: UDP, Principles of Reliable Data Transfer, Connection-Oriented Transport: TCP, Flow Control, Congestion Control.

Unit – V **(9)**

Application Layer-Principles of Network Applications, The Web and HTTP, File Transfer: FTP, Electronic Mail in the Internet, Domain Name system (DNS), Peer-to-Peer Applications. Audio and video streaming over UDP, HTTP. Adaptive streaming. Voice over IP.

Textbooks:

1. J.F. Kurose and K.F. Ross, Computer networking: a top-down approach, Pearson
2. Larry L.Peterson, Peter S. Davie, "Computer Networks", Elsevier, Fifth Edition, 2012.

Reference Books:

1. Computer Networks – Third Edition – Andrew S. Tanenbaum, Prentice Hall of India.
2. U. Black, "Computer Networks-Protocols, Standards and Interfaces", PHI, 1996.
3. Laura Chappell, "Introduction to Cisco Router Configuration", Techmedia, 1999.
4. Michael A. Miller, "Data & Network Communications", Vikas Publication, 1998.
5. William A. Shay, "Understanding Data Communications & Networks", Vikas Publication, 1999.
6. W. Stallings, "Computer Communication Networks", PHI, 1999.

Course Code: **CSE312C**	**Python Programming**	Credits: **03** L – **3** P – **0**

Course Outcomes (COs):

- Gain a solid understanding of Python programming language basics.
- Demonstrate proficiency in using control structures.
- Acquire the skills to manipulate common data structures in Python.
- Understand and apply the principles of Object-Oriented Programming (OOP).
- Gain exposure to Python libraries and frameworks for practical applications.

Unit – I **(8)**

Introduction to Python: Why Python?, History, Features, and Applications, Python 2 vs. Python 3, Setting up Python Environment; Installation of Python, Introduction to IDEs (Integrated Development Environments), Python Basics; Variables, Data Types, and Operators, Basic Input and Output.

Unit – II **(8)**

Control Flow and Functions: Control Structures; Conditional Statements (if, elif, else), Loops (for, while), Functions; Defining Functions, Parameters and Return Values, Scope and Lifetime of Variables, Exception Handling; Handling Errors with Try-Except Blocks.

Unit – III **(8)**

Data Structures in Python: Lists and Tuples; Operations and Methods, Dictionaries and Sets; Key-Value Pairs and Uniqueness, String Manipulation; String Operations and Methods.

Unit – IV **(8)**

Object Oriented Programming (OOP): Introduction to OOP; Classes and Objects, Inheritance and Polymorphism; Code Reusability and Flexibility, File Handling; Reading and Writing Files in Python.

Unit – V **(8)**

Python Libraries and Applications: Overview of Python Libraries; NumPy, Pandas, Matplotlib, Introduction to Web Development with Flask; Basics of Flask Framework, Introduction to Data Science with Python; Basic Concepts and Applications.

Textbooks:

1. "Python Cookbook" by David Beazley and Brian K. Jones
2. "Fluent Python" by Luciano Ramalho

Reference Books:

1. Python for Data Analysis" by Wes McKinney.
2. "Automate the Boring Stuff with Python" by Al Sweigart
3. "Python Crash Course" by Eric Matthes.

Course Code: **CSE313C**	**Microprocessors, Peripherals and Interfacing**	Credits: **03** L – **3** P – **0**

Course Outcomes (COs):

- To have a complete knowledge of architecture of micro-processor 8085, its pin layout and description of all signals.
- To understand the instruction cycle, timing diagrams, registers, flags etc. of 8085 micro-processor in detail.
- To have a knowledge of instructions, addressing modes, interrupts, subroutines, conditional call instructions of 8085 and be able to perform assembly language programming for numerous operations in 8085.

Unit – I **(8)**

Microcomputer Structure and Operations: Basic Microcomputer Elements, Microprocessors and Memory: Typical 8, 16 and 32 bit Microprocessors, 8085 Microprocessor - Memory Technology, Pin Description and Internal Architecture of 8085, registers, flags, buses. Architecture of 8-bit Microprocessor: Instruction Classification: 1-byte, 2-byte and 3-byte instructions.

Unit – II **(11)**

Assembly Language Programming: Programming Model of 8085, Registers, Fetch, Execute Operation of CPU, Instruction Set, Addressing Modes, Basic Operations, Microprocessor Arithmetic, Program Flow, Control Using Looping and Branching. Concepts of instruction cycle, machine cycle, and t-states. Opcode fetch machine cycle, memory read/write machine cycles, I/O read/write machine cycles, time delays.

Unit – III **(9)**

Counters, time delays and related examples. Stacks and subroutines. Interrupts: Interrupt structure of 8085 microprocessor, vectored and non-vectored interrupts, Priority Management, Address Decoding.

Unit – IV **(10)**

Microprocessors Interfacing: Interfacing concepts, Parallel Input Output, Memory mapped I/O, Memory Interfacing, Direct Memory Access, 8237 DMA Controller. The Serial Subsystems. 8255 Programmable Peripheral Interface, Analog Converter Subsystem.

Unit – V **(8)**

Introduction to 8086 architecture: Pin Configuration, Main features and addressing modes, difference between 8085 and 8086. Introduction of 8088, 80186, 80386 microprocessors.

Textbooks:

1. Microprocessor Architecture, Programming, and Applications with the 8085 –Ramesh S. Gaonkar, Pub: Penram International.
2. A.K. Ray and K.M. Bhurchandi Advanced Microprocessors and Peripherals, third Edition, Tata McGraw Hill, 2012. 2. Barry B Bray, The Intel Microprocessor 8086/8088, 80186,80286, 80386 and 80486 Arcitecture, programming and interfacing, PHI, 8th Edition, 2009.
3. Hall D.V. "Microprocessor and Interfacing-Programming and Hardware", 2nd Ed., Tata McGraw-Hill Publishing Company Limited, 2008.

Reference Book:

1. 8085 Microprocessor and its Applications, by A. Nagoor Kani, Third Edition, TMH Education Pvt. Ltd.
2. Mohamed Rafiquazzaman, Microprocessor and Microcomputer based system design, Universal Book stall, New Delhi.

Course Code: **CSE314C**	**Probability and Statistics**	Credits: **03** L – **3** P – **0**

Course Outcomes (COs):

- Solve basic probability problems.
- Understand the random variables and various related concepts like joint, conditional and marginal probabilities.
- Identify and work with different distributions
- Appreciate the common ground between probability and statistics.
- Solve basic probability problems.

Unit – I **(9)**

Statistics: Measures of central tendency and Measures of variations (Dispersions), Moments, Measures of Skewness and Kurtosis. Moment generating functions, problems.

Unit – II **(9)**

Probability: Random experiment, sample space, events, classical, statistical and axiomatic definitions of probability. Statements and proof of theorems on addition and multiplication of probabilities.

Unit – III **(9)**

Conditional Probability: Bayes theorem on conditional probability. Random variables, Derivation of formulae for mean, variance and moments of random variables for discrete and continuous cases. Laws of expectation.

Unit – IV **(9)**

Standard Distributions: Bernoulli and Binomial distributions, Poisson and Normal Distributions, Beta and Gamma Distribution, t-Distribution, F-Distribution, Chi-square Distribution. Central Limit Theorem.

Unit – V **(9)**

Method of Least Squares & Correlation: Methods of least squares, fitting of straight line and parabola of degree 'p'. Regression and Correlation. Multiple and Partial Correlation Problems.

Textbooks:

1. Fundamentals of Mathematical Statistics, by S.C.Gupta and V.K. Kapoor, Sulltan Chand & Sons New Delhi, Latest edition.
2. Statistical Theory and Methodology in Science & Engineering, by Brownlee, John Wiley & Sons.

Reference Books:

1. Introduction to Mathematical Statistics, by R.E.Walpole 3rd edition New York Macmillan publication.
2. Data Analysis for Scientists & Engineers, by Meyer, John Wiley & Sons.

Course Code: **CSE315C**	**Operating System Lab**	Credits: **01** L – **0** P – **2**

Course Outcomes (COs):
By the end of this course, students will be able to:

- Execute essential Linux commands for file system navigation, user management, and process control.
- Develop and execute shell scripts for automating tasks in a Linux environment.
- Implement system-level programs using C to perform file handling and process management using system calls.
- Perform basic Linux system administration tasks such as user/group management, software installation, and job scheduling.
- Monitor system performance and demonstrate basic networking and troubleshooting commands in Linux.

List of Experiments

1. Execute basic Linux commands for file and directory manipulation, and explore the Linux file system hierarchy.
2. Demonstrate file permission management using *chmod*, *chown*, and edit files using *vim* or *nano* editor.
3. Write a shell script to demonstrate the use of variables, user input, and command-line arguments.
4. Write a shell script using conditionals and loops to automate tasks like file checks or menu-based utilities.
5. Manage processes using Linux commands like *ps*, *top*, *kill*, *nice*, *fg*, and *bg*.
6. Implement a C program to create a child process using *fork()* and demonstrate the use of *exec()* and *wait()* system calls.
7. Write a C program using file system calls *[open(), read(), write(), close()]* to perform basic file operations.
8. Perform user and group management in Linux using commands such as *useradd, passwd, groupadd,* and *sudo*.
9. Install and remove software using package management tools (*apt, dpkg*) and schedule tasks using *cron* and *at*.
10. Monitor system performance using commands like *free, df, du, top*, and demonstrate basic networking using *ping, ip*, and *traceroute.*

Textbook:

1. Silberschatz, A., Galvin, P. B., & Gagne, G., Operating System Concepts, Wiley.

Reference Books:

1. Neil Matthew & Richard Stones, Beginning Linux Programming (4th Edition), Wrox Press.
2. M. G. Venkateshmurthy, Introduction to UNIX and Shell Programming, Pearson Education.
3. Yashavant Kanetkar, Let Us Linux, BPB Publications.
4. W. Richard Stevens & Stephen A. Rago, Advanced Programming in the UNIX Environment, Addison-Wesley.

Online Resources:

1. Linux Command and Shell Scripting: https://linuxcommand.org
2. https://tldp.org
3. https://explainshell.com
4. https://man7.org/linux/man-pages/
5. https://www.geeksforgeeks.org
6. https://www.tutorialspoint.com/unix

Course Code: **CSE316C**	**Computer Networks Lab**	Credits: **01** L – **0** P – **2**

Course Outcomes (COs):

- Understanding of the design, troubleshooting, modeling and evaluation of computer networks
- Identify and use various networking components.
- Understand different transmission media and design cables for establishing a network.
- Implement any topology using network devices.
- Compare routing algorithms.

List of Experiments

1. Study of different types of network cables and tools
2. Implement the cross-wired cable and straight through cable using clamping tool.
3. Study of Network Devices in Detail.
4. Concept of Network IP Address
5. Creating point-to-point network and sharing files
6. Building a Local Area Network.
7. Study of basic network command and Network configuration commands.
8. Study of basic network command and Network configuration commands.
9. Configure a Network topology using packet tracer software.
10. Configure a Network topology using packet tracer software.
11. Implementation of Static Routing using Packet Tracer software.
12. Configure a Network using Distance Vector Routing protocol.
    Routing Information Protocol (RIP)
13. Configure Network using Link State Vector Routing protocol.
    Open Shortest Path First (OSPF)
14. Implementation of a VLAN using Packet Tracer
15. Implementation of a VLAN using Packet Tracer

Preferred Tool:

- Packet Tracer

Textbooks:

1. J.F. Kurose and K.F. Ross, Computer networking: a top-down approach, Pearson
2. Larry L.Peterson, Peter S. Davie, “Computer Networks”, Elsevier, Fifth Edition, 2012.

Reference Books:

1. Computer Networks – Third Edition – Andrew S. Tanenbaum, Prentice Hall of India.
2. U. Black, “Computer Networks-Protocols, Standards and Interfaces”, PHI, 1996.
3. Laura Chappell, “Introduction to Cisco Router Configuration”, Techmedia, 1999.
4. Michael A. Miller, “Data & Network Communications”, Vikas Publication, 1998.
5. William A. Shay, “Understanding Data Communications & Networks”, Vikas Publication, 1999.
6. W. Stallings, “Computer Communication Networks”, PHI, 1999.
7. William A. Shay, “Understanding Data Communications & Networks”, Vikas Publication, 1999.
8. W. Stallings, “Computer Communication Networks”, PHI, 1999.

Course Code: **CSE317C**	**Python Programming Lab**	Credits: **01** L – **0** P – **2**

Course Outcomes (COs):

- Gain a solid understanding of Python programming language basics.
- Demonstrate proficiency in using control structures.
- Acquire the skills to manipulate common data structures in Python.
- Understand and apply the principles of Object-Oriented Programming (OOP).
- Gain exposure to Python libraries and frameworks for practical applications.

List of Experiments

1. Setting up Python Environment:
   - Verify the installation by running a simple Python script.
   - Introduction to Python interpreter and interactive mode.
2. Introduction to IDEs:
   - Install an Integrated Development Environment (IDE) such as PyCharm, Visual Studio Code, or IDLE.
   - Explore the features and interface of the chosen IDE.
   - Create and run a Python script within the IDE.
3. Python Basics:
   - Practice defining variables and assigning values of different data types (integers, floats, strings, booleans). Perform arithmetic operations using Python operators.
   - Use print() function for basic input and output operations.
4. Control Flow and Functions:
   - Write Python code to implement conditional statements (if, elif, else) for decision-making.
   - Create loops (for, while) to iterate over data structures or perform repetitive tasks.
   - Define and call functions with parameters and return values.
   - Demonstrate understanding of variable scope and lifetime.
5. Exception Handling:
   - Write Python code to handle errors using try-except blocks.
   - Handle specific exceptions and provide appropriate error messages.
   - Test exception handling by intentionally raising errors.
   - Perform string manipulation tasks using string operations and methods.
6. Data Structures in Python:
   - Create and manipulate lists and tuples using built-in methods and operations.
   - Explore dictionaries and sets, understanding their properties and methods.
7. Object Oriented Programming (OOP):
   - Define classes and create objects in Python.
   - Implement inheritance and polymorphism concepts to demonstrate code reusability.
   - Showcase file handling by reading from and writing to files using Python.
8. Python Libraries and Applications:
   - Explore popular Python libraries such as NumPy, Pandas, and Matplotlib.
   - Import and use functions from these libraries to perform basic data analysis and visualization tasks.
   - Learn the basics of Flask framework for web development, including routing and rendering templates.
10. Introduction to Data Science with Python:
    - Explore basic concepts of data science and its applications.
    - Perform data manipulation and analysis using Pandas library.
    - Visualize data using Matplotlib for creating plots and charts.
11. Introduction to Data Science with Python:
    - Explore basic concepts of data science and its applications.
    - Perform data manipulation and analysis using Pandas library.
    - Visualize data using Matplotlib for creating plots and charts.

Textbooks:

1. Eric Matthes, Python Crash Course, No Starch Press, 2nd Edition.

Reference Books:

1. Wes McKinney, Python for Data Analysis, O'Reilly Media.
2. Al Sweigart, Automate the Boring Stuff with Python, No Starch Press.
3. Mark Lutz, Learning Python, O'Reilly Media.

Online Resources:

1. https://www.python.org
2. https://www.w3schools.com/python/
3. https://developers.google.com/edu/python
4. https://www.kaggle.com/learn/python
5. https://www.youtube.com/c/CoreySchafer

Course Code: **CSE318C**	**Microprocessors, Peripherals and Interfacing Lab**	Credits: **01** L – **0** P – **2**

Course Outcomes (COs):

- By the end of this laboratory course, students will be able to:
- Develop and execute assembly language programs using the 8085 instruction set for performing basic arithmetic, logical, and control operations.
- Analyze and implement algorithms for data manipulation tasks such as searching, sorting, and code conversion using 8085 microprocessor.
- Demonstrate proficiency in memory and register operations including stack usage, subroutine calls, and block data transfer.
- Interface simple I/O devices like LEDs and 7-segment displays with the microprocessor system (or simulator), and control them through software.
- Apply microprocessor programming concepts in solving real-time problems through structured, modular, and optimized assembly code.

List of Experiments

1. Write assembly language programs to perform
    a. 8-bit addition and subtraction
    b. 16-bit addition and subtraction using register pairs
2. Write assembly language programs to perform
    a. Multiplication of two 8-bit numbers using repeated addition
    b. Division of an 8-bit number by another using repeated subtraction
3. Perform bitwise AND, OR, XOR on two numbers.
4. Transfer a block of N bytes from one memory location to another and exchange two blocks.
5. Search the largest and smallest number from a list of N numbers.
6. Write a program to sort N numbers using Bubble Sort.
7. Count the number of even and odd elements from a list.
8. Write assembly language programs to perform
    a. Binary to BCD and BCD to Binary
    b. Binary to ASCII and ASCII to Binary
9. Use PUSH, POP instructions
10. Implement subroutines using CALL and RET
11. Display digits 0–9 on a 7-segment display
12. Create a binary counter output using LEDs

Textbook:

1. Ramesh S. Gaonkar, Microprocessor Architecture, Programming and Applications with the 8085, Penram International Publishing.

Reference Books:

1. A.K. Ray and K.M. Bhurchandi, Advanced Microprocessors and Peripherals, McGraw Hill Education.
2. Vijay Goel & Himanshu Goel, 8085 Microprocessor Programming and Interfacing, Pearson Education.
3. Douglas V. Hall, Microprocessors and Interfacing: Programming and Hardware, McGraw Hill.

Online Resources:

1. NPTEL Online Course, Microprocessors and Microcontrollers by Prof. Santanu Chattopadhyay, IIT Kharagpur
2. GNUSim8085 Simulator: https://sourceforge.net/projects/gnusim8085/
3. EMU8085 Emulator: http://www.emu8085.net/
4. https://www.tutorialspoint.com/microprocessor/microprocessor_8085_architecture.htm
5. https://www.electronicshub.org/8085-microprocessor-programs/