OPERATING SYSTEM
(CS401)

Objective: To give knowledge about principles of modern operating system.	
Unit	**Topic**
I	**Introduction to Operating Systems:** Role and purpose of operating systems, Operating System Services, Classification of Operating systems, Operating System Structure, System Calls.
II	**CPU Scheduling:** Process vs. Program, Process States, Process Transition Diagram, Process Control Block, Process Address Space, Schedulers, Scheduling Concepts, Performance Criteria, Scheduling Algorithms, Threads, Deadlock Problem, Deadlock Characterization, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock, Methods for Deadlock Handling.
III	**Inter-process Communication:** Race conditions, Critical sections, Mutual exclusion, Critical-section problem, Algorithmic approach to implementing critical sections, Hardware support for process synchronization, Semaphores, Mutexes, Monitors. **Classic Problems of Synchronization:** Producers-consumers with bounded buffers problem, Readers-writers problem, Dining-philosophers problem
IV	**Memory Management:** Introduction, Logical and Physical Address Space, Swapping, Contiguous Memory Allocation, Fragmentation, Paging, Structure of Page Table, Segmentation, Segmentation with Paging. Virtual Memory: Demand Paging, Performance of Demand Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing
V	**File system:** File Concept, Access Methods, Directories, Mounting of File-System, File-System Structure, File-System Implementation, Allocation Methods. I/O Devices, and I/O Subsystems, I/O Buffering, Disk Storage and Disk Scheduling, Disk Management, RAID.

References:

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts, John Wiley, 2008,
2. William Stallings, Operating Systems: Internals and Design Principles. PHI, 2012.
3. AS Tanenbaum, Modern Operating Systems, Pearson, 2015.
4. AS Tanenbaum, AS Woodhull, Operating Systems Design and Implementation, PHI, 2006.

COMPUTER GRAPHICS
(CS402)

Objective: The goal of subject is to provide the students a broad exposure to the computer graphics field in order to prepare for follow-on study.	
Unit	**Topic**
I	**Introduction and Line Generation:** Types of computer graphics, Graphic Displays Random scan displays, Raster scan displays, Frame buffer and video controller, Points and lines, Line drawing algorithms, Circle generating algorithms, Mid-point circle generating algorithm, and parallel version of these algorithms
II	**Transformations:** Basic transformation, Matrix representations and homogenous coordinates, Composite transformations, Reflections and shearing. **Windowing and Clipping:** Viewing pipeline, Viewing transformations, 2-D Clipping algorithms- Line clipping algorithms such as Cohen Sutherland line clipping algorithm, Liang Barsky algorithm, Line clipping against non rectangular clip windows; Polygon clipping – Sutherland Hodgeman polygon clipping, Weiler and Atherton polygon clipping,Curve clipping, Text clipping
III	**Three Dimensional:** 3-D Geometric Primitives, 3-D Object representation, 3-D Transformation, 3-D viewing, projections, 3-D Clipping.
IV	**Curves and Surfaces**: Quadric surfaces, Spheres, Ellipsoid, Blobby objects, Introductory concepts of Spline, Bspline and Bezier curves and surfaces.
V	**Hidden Lines and Surfaces:** Back Face Detection algorithm, Depth buffer method, Abuffer method, Scan line method, basic illumination models– Ambient light, Diffuse reflection, Specular reflection and Phong model, Combined approach, Warn model, Intensity Attenuation, Color consideration, Transparency and Shadows.

References:

1. Donald Hearn and M. Pauline Baker, “ Computer Graphics”, PHI, 2016.
2. Steven Harrington, “Computer Graphics: A Programming Approach”, TMH, 1987.
3. Foley James D, “Computer Graphics”, AW, 2014

WEB TECHNOLOGY
(CS403)

Objective: To provide a basic understanding about web world and develop web sites based projects.	
Unit	**Topic**
I	**Introduction to Web Technology**: Internet, WWW, Web Browsers with suitable examples, Web Servers with suitable examples, URL, HTTP, MIME. Introduction to HTML& DHTML : Basic Syntax, HTML Document Structure , Text Formatting, Images, Lists, Links, Tables, Frames, Forms. Cascade Style Sheets**:** Levels Of Style Sheets, Specification Formats, Style Classes, Properties, Colors, Span and Div tags.
II	**Introduction to Java Script:** Overview of java Script, Syntactic characteristics, Primitives, Operator and Expression, control statements, Arrays, functions, errors in scripts, Document Object Model(DOM),Event driven computation, element access in Java script, The navigator Object. Dynamic Document with Java Script : Element positioning, Moving elements, Changing colors and fonts, Dynamic content, Locating the mouse Cursor, Slow movements of elements, Dragging and Dropping Elements.
III	**Introduction to XML:** Syntax of XML, Document Structure, Document type definition, Namespaces,XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX.
IV	**Introduction to PHP:** Overview of PHP, general server characteristics, Creating PHP Pages, Form handling , Data Base access with PHP & MySql. JSP & Servlets: Basics of JSP, Tags, Session Handling, Redirection and Basics of Servlet
V	**Database Connectivity:** Connection to various databases (MySQL, Oracle). Database connectivity in PHP, JSP, Other server languages, Basic SQL Queries and Statements.

References:

1. Programming world wide web- Robert W.Sebesta , Pearson, 2008.
2. Beginners PHP, Apache, MY Sql, Web Development, by Michael Glass, Wrox, 2004.

COMPUTER ORGANIZATION &ARCHITECTURE
(CS404)

Unit	Topic
I	**Functional blocks of a computer**: CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU–registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Case study – instruction sets of some common CPUs. **Data representation**: 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.
II	**Introduction** to x86 architecture. **CPU control unit design**: hardwired and micro-programmed design approaches, Case study – design of a simple hypothetical CPU. **Memory system design**: semiconductor memory technologies, memory organization.
III	**Peripheral devices and their characteristics**: Input-output subsystems, I/O device interface, I/O transfers–program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes-role of interrupts in process state transitions, I/O device interfaces - SCII, USB
IV	**Pipelining**: Basic concepts of pipelining, throughput and speedup, pipeline hazards. **Parallel Processors**: Introduction to parallel processors, Concurrent access to memory and cache coherency.
V	**Memory organization:** Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.

Text Book (s):

1. “Computer Organization & Architecture”, Rajaraman, PHI Learning
2. “Computer Organization and Embedded Systems”, 6th Edition by Carl Hamacher, McGraw Hill Higher Education.

OPERATING SYSTEM LAB
(CS451)

LIST OF EXPERIMENTS

1. Write a program to implement fork system call.
2. Write a program to implement critical section problems.
3. Write a program to implement classical problems of process synchronization.
4. Write a program to implement non-preemptive scheduling algorithm
5. Write a Program to implement preemptive scheduling algorithm.
6. Write a program to Implement Banker's algorithm.
7. Write a program to implement page replacement algorithms.
8. Write a program to implement Disk Scheduling algorithms.
9. Write a Program to implement file allocation methods.
10. Write a Program for MFT and MVT first fit and best fit.

COMPUTER GRAPHICS LAB
(CS452)

LIST OF EXPERIMENTS

1. Write a program to animate a bouncing ball using in built functions.
2. Write programs to draw Line, circle using DDA technique and polynomial equation.
3. Write programs to implement Mid point line, circle.
4. Write programs to implement Bresenham's line, circle.
5. Write programs to implement Flood-fill , boundary fill algorithm.
6. Write a program to implement scan line polygon fill algorithm.
7. Write a program to implement all the basic 2D-Transformations.
8. Write a program to implement cohen-sutherland line clipping algorithm.
9. Write a Program to implement sutherland-Hodgmn clipping algorithm.
10. Write a Program to implement Bezier curve.

WEB TECHNOLOGY LAB
(CS453)

LIST OF EXPERIMENTS

1. Write a program in HTML to display different styles of heading text.
2. Write a program to display the processes to be followed for a patient when he enters for a complete check-up. Use ordered lists and unordered lists.
3. Write a program to display a traditional Newspaper with the use of table tags.
4. With the help of “IMAGE” tags write a program to display the image along with some contents.
5. Use “Anchor” tag to write a program for displaying various Menus.
6. Use mapping technique, to map a particular part of image and move the control corresponding to that area. For e.g., in an image, if there are bat, ball, stump etc. When you click stump control should move to a file call St.htm.
7. Create frames that have details about various cities.
8. Create a form to display the kinds of food available in a Restaurant. (Use checkboxes wherever necessary)
9. Write a program to “reload” a page automatically once in 5 seconds.
10. Write a program using CSS to set the background colour, font, and paragraph.

COMPUTER ORGANIZATION &ARCHITECTURE LAB

(CS454)

LIST OF EXPERIMENTS

1. Implementing HALF ADDER, FULL ADDER using basic logic gates
2. Implementing Binary -to -Gray, Gray -to -Binary code conversions.
3. Implementing 3-8 line DECODER.
4. Implementing 4x1 and 8x1 MULTIPLEXERS.
5. Verify the excitation tables of various FLIP-FLOPS.
6. Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
7. Design of an 8-bit ARITHMETIC LOGIC UNIT.
8. Design the data path of a computer from its register transfer language description.
9. Design the control unit of a computer using either hardwiring or microprogramming based on its register transfer language description.
10. Implement a simple instruction set computer with a control unit and a data path.

COMPUTER NETWORKS
(CS601)

Objective: The objective of this course is to provide basic exposure to computer networks theory and implementations.

Unit	Topic
I	**Introduction:** Networks, Internet, Network Components, Network Categories, Applications of Computer Networks **Reference Models:** Concept of Layering, OSI Model, TCP/IP Protocol Suite, Functions of Layers **Physical Layer:** Transmission Mode, Physical Topology, Multiplexing, Transmission Media, Switching
II	**Data Link Layer:** Design Issues, Error Detection and Correction Techniques, Elementary Data Link Protocols, Sliding Window Protocols, Multiple Access Protocols, Ethernet, Connecting Devices
III	**Network Layer:** Logical addressing, IPv4 Addresses, NAT, IPv6 Addresses, Internet Protocol, IPv4, IPv6, Internetworking, Internet Control Protocols, Routing Algorithms, Distance Vector Routing, Link State Routing, Routing in the Internet
IV	**Transport Layer:** Process-to-Process Delivery, Transport Layer Protocols, UDP, User Datagram, TCP, TCP Segment, TCP Connection, Flow Control and Error Control, TCP Transmission Policy, Principles of Congestion Control, TCP Congestion Control, Quality of Service.
V	**Application Layer:** Principles of Network Applications, WWW and HTTP, Non-Persistent and Persistent Connections, Cookies, Web Caching, File Transfer, Remote Logging, Electronic Mail in the Internet, Domain Name System, Security: Introduction, Cryptography and Cryptanalysis, Public Key Cryptography Algorithms, RSA Algorithm, DES, Authentication and Authorization

References:

1. AS Tanenbaum, DJ Wetherall, Computer Networks, Prentice-Hall, 2010. .
2. LL Peterson, BS Davie, Computer Networks: A Systems Approach, Morgan-Kauffman, 2011.
3. W Stallings, Cryptography and Network Security, Principles and Practice, Prentice-Hall, 2005.

COMILER DESIGN
(CS602)

<table>
<tr><td colspan="2"><b>Objective:</b> Students will have a fair understanding of some standard passes in a general-purpose compiler.</td></tr>
<tr><th>Unit</th><th>Topic</th></tr>
<tr><td><b>I</b></td><td><b>Introduction to Compiler</b>: Phases and passes, Bootstrapping, Finite state machines and regular expressions and their applications to lexical analysis, implementation of lexical analyzers,, LEX-compiler, Formal grammars and their application to syntax analysis,, ambiguity, The syntactic specification of programming languages: Context free grammars, derivation and parse trees, capabilitiesofCFG.</td></tr>
<tr><td><b>II</b></td><td><b>Basic Parsing Techniques</b>: Parsers, Shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR(0) items, constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, YACC tool.</td></tr>
<tr><td><b>III</b></td><td><b>Syntax-directed Translation:</b> Syntax-directed Translation schemes, Intermediate code, postfix notation, Parse trees & syntax trees, three address code, quadruple & triples, Translation of simple statements and control flow statements, Type checking, Type conversions, Equivalence of type expressions, Overloading of functions and operations.</td></tr>
<tr><td><b>IV</b></td><td><b>Symbol Tables</b>: Data structure for symbols tables, representing scope information.
<b>Run-TimeAdministration:</b> Implementation of simple stack allocation scheme, storage allocation in block structured language. Error Detection & Recovery: Lexical Phase errors, syntactic phase errors semantic errors.</td></tr>
<tr><td><b>V</b></td><td><b>Code Generation</b>: Design Issues, the Target Language. Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, Code Generator. Code optimization: Machine-Independent Optimizations, Loop optimization, DAG representation of basic blocks, value numbers and algebraic laws, Global Data-Flow analysis.</td></tr>
</table>

References:

1. K.D. Cooper, and Linda Torczon, Engineering a Compiler, Morgan Kaufmann, 2011.
2, K.C. Louden, Compiler Construction: Principles and Practice, Cengage Learning, 1997.
3. D. Brown, J. Levine, and T. Mason, LEX and YACC, O‟Reilly Media, 1992.

ARTIFICIAL INTELLIGENCE
(CS603)

<table>
<tr><td colspan="2"><b>Objective:</b> To learn the concepts of Artificial Intelligence and the methods of solving problems using Artificial Intelligence.</td></tr>
<tr><th>Unit</th><th>Topic</th></tr>
<tr><td><b>I</b></td><td><b>Introduction:</b> Introduction to Artificial Intelligence, Foundations and History of Artificial Intelligence , Application of Artificial Intelligence
Communication - Communication among agents, natural language processing, formal grammar, parsing, grammar</td></tr>
<tr><td><b>II</b></td><td><b>Introduction to Search</b>: Searching for solutions, Uniformed search strategies, Informed search Strategies, Local search algorithms and optimistic problems, Adversarial Search, Search for games, Alpha - Beta pruning.</td></tr>
<tr><td><b>III</b></td><td><b>Knowledge Representation & Reasoning:</b> Propositional logic, Theory of first order logic, Inference in First order logic, Forward & Backward chaining, Resolution, Probabilistic reasoning, Utility theory, Hidden Markov Models (HMM), Bayesian Networks.</td></tr>
<tr><td><b>IV</b></td><td><b>Decision making-</b> Utility theory, utility functions, and Decision theoretic Expert systems. Default reasoning, Fuzzy sets and fuzzy logic; AI languages and tools - Lisp, Prolog,</td></tr>
<tr><td><b>V</b></td><td><b>Pattern Recognition :</b> Introduction, Design principles of pattern recognition system, Statistical Pattern recognition, Parameter estimation methods - Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA),</td></tr>
</table>

References:

1. Kevin Night and Elaine Rich, Nair B., “Artificial Intelligence (SIE)”, McGraw Hill, 2008.
2. Dan W. Patterson, “Introduction to AI and ES”, Pearson Education, 2007.
3. Peter Jackson, “Introduction to Expert Systems”, Pearson Education, 2011.

Python Programming
(CS604)

Objective: To familiarize the students with advanced databases and techniques of retrieving and storing information.	
Unit	**Topic**
I	**Introduction To Python:** Installation and Working with Python Understanding Python variables Python basic Operators Understanding python blocks **Values and Variables :** Integer and String Values, Identifiers, User Input, String Formatting, Expressions and Arithmetic Examples
II	**Python Data Types:** Declaring and using Numeric data types: int, float, complex Using string data type and string operations Defining list and list slicing Use of Tuple data type
III	**Python Conditional Statements and looping:** If, If- else, Nested if-else For, While Nested loops
IV	**Python String, List And Dictionary Manipulations:** Building blocks of python programs, Understanding string in build methods, List manipulation using in build methods ,Dictionary manipulation Programming using string, list and dictionary in build functions
V	**Python Object Oriented Programming:** Oops Concept of class, object and instances Constructor, class attributes and destructors ,Real time use of class in live projects ,Inheritance , overlapping and overloading operators Adding and retrieving dynamic attributes of classes Programming using Oops support..

References:

Chun, J Wesley, Core Python Programming, Second Edition, Pearson, 2007 Reprint 2010

Essential Reading / Recommended Reading

[1] Barry, Paul, Head First Python, 2nd Edition, O Rielly, 2010

[2] Lutz, Mark, Learning Python, 4th Edition, O Rielly, 2009

COMPUTER NETWORKS LAB
(CS651)

LIST OF EXPERIMENTS

1. To learn basics of the packet tracer simulator tool.
2. Write a program in C to implement bit stuffing and character stuffing.
3. To connect the computers in Local Area Network and to detect collision of packets.
4. To configure DHCP and DNS server for a given network in packet tracer simulator tool.
5. Write a C program to get the MAC or Physical address of the system using ARP (Address Resolution Protocol) and to subnet a given network according to the requirements in packet tracer simulator tool. .
6. To configure router using command line. Also observe the datagram formats in packet tracer simulator tool.
7. To configure NAT for a given network in packet tracer simulator tool.
8. Write a program to implement TCP & UDP Sockets.
9. Write a C program to transmit a character, a string and a file from one computer to another using RS-232 cable and to configure static routing in packet tracer simulator tool.
10. To configure dynamic routing protocols in packet tracer simulator tool.

COMILER DESIGN LAB
(CS652)

LIST OF EXPERIMENTS

1. Write a program to check whether a string belongs to the grammar or not.
2. Practice of Lex of Compiler writing.
3. Write a LEX program to count number of printf and scanf from a given c program file and replace them with write and read respectively.
4. Write a program to check whether a grammar is left recursive and remove left recursion.
5. Write a program to remove left factoring
6. Write a program to compute FIRST and FOLLOW of non-terminals.
7. Write a program to check whether a grammar is Operator precedent. .
8. Practice of Yacc of Compiler writing.
9. Write a YACC program to recognize the grammer[a^nb/n>0] . Test whether the following string belongs to this grammer.
10. Write a YACC & LEX program to identify valid if and if-else statement.

Python Programming LAB
(CS653)

LIST OF EXPERIMENTS

1. Implement a sequential search
2. Create a calculator program
3. Explore string functions
4. Implement Selection Sort
5. Implement Stack
6. Read and write into a file
7. Demonstrate usage of basic regular expression
8. Demonstrate use of List
9. Demonstrate use of Dictionaries