

(VI Semester)

(CS1601) Internet Fundamentals & Application (2 -1 - 0)

Evolution of Internet, TCP/IP addressing and routing. Internet applications: FTP, Telnet, Email, Chat. World Wide web, HTTP protocol. Designing web pages: HTML, forms, CGI scripts and clickable maps, JavaScript.

Java servlets, Perl, DHTML, XML. E-Commerce and security issues including symmetric and asymmetric key, encryption and digital signature authentication. Internet telephony, virtual reality over the web. Intranet and extra net, firewall design issues.

Suggested Text Books & References

- * Black, "Computer Networks".
- * Stevens, "Unix Networking Programming", 2nd Edition

(CS1602) Interactive Computer Graphics (2-1-0)

Graphics hardware and display devices; graphics primitives- drawing lines and curves; 2d and 3d transformations segments and their applications; generating curves, surfaces and volumes in 3d, wire-frame models, Bezier and spline curves and surfaces; geometric modeling- elementary geometric algorithms for polygons, boundary representations, constructive solid geometry, spatial data structures; hidden surface and line elimination; rendering, shading, light models. Realistic image synthesis techniques, textures and image-based rendering; video games and computer animation.

Laboratory: Programming for generating lines, curves and rendered surfaces. Interactive graphics programming- modeling and updating objects in an object hierarchy, video games, computer animation and realistic image synthesis.

Suggested Text Books & References

- Rogers "Procedural Elements of Computer Graphics", McGraw Hill.
- Newman & Sproule, "Principles of Interactive Computer Graphics", ", McGraw Hill.1987
- Harringtons. S., "Computer Graphics", A Programming Approach 2nd Edition, McGraw Hill. 1987.
- Rogers & Adams "Mathematical Elements of computer Graphics", 2nd Edition, McGraw Hill.
- Henry Baper,"Computer Graphics".

(CS1603) LANGUAGE PROCESSORS (3-1-0)

Compiler Structure: Analysis - Synthesis model of compilation, various phases of a compiler, Tool based approach to compiler construction.

Lexical Analysis: Interface with input, parser and symbol table, Token, lexeme and patterns. Difficulties in lexical analysis, Error reporting, Implementation, Regular definition, Transition diagrams, LEX.

Syntax Analysis: CFGs, Ambiguity, associativity, precedence, Top down parsing, Recursive descent parsing, Transformation on the grammars, Predictive parsing, Bottom up parsing, Operator precedence grammars, LR parses (SLR, LALR, LR), YACC.

Syntax Directed Definitions: Inherited and synthesized attributes, dependency graph, Evaluation order, bottom up and top down evaluation of attributes, L- and S- Attributes definitions.

Type Checking: Type system, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions.

Run Time System: Storage organization, activation tree, activation record, parameter passing, symbol table, dynamic storage allocation.

Intermediate Code Generation: Intermediate representations, translation of declarations, assignments, control flow, Boolean expressions and procedure calls. Implementation issues.

Code Generation and Instruction selection: Issues, basic blocks and flow graphs, register allocation, code generation, Dag representation of programs, code generation from dags, peep hole optimization.

Suggested Text Books & References

- Aho, A. V. Sethi R. and Ullman, J.D. "Compilers Principles, Techniques and Tools", Addison-Wesley, 1988.
- Fischer, C. and LeBlanc, R.J. "Crafting a Compiler with C, Benjamin Cummings", 1991.
- Holub, A.C. "Compiler Design in C", Prentice Hall of India, 1993.

(CS1604) Computer Architecture (3-1-0)

Overview of von Neumann architecture: Instruction set architecture; The Arithmetic and Logic Unit, The Control Unit, Memory and I/O Devices and their interfacing to the CPU; Measuring and reporting performance ; CISC and RISC processors, Pipelining, Basic concepts of pipelining, data hazards, control hazards, and structural hazards; Techniques for overcoming or reducing the effects of various hazards. Hierarchical Memory Technology: Inclusion, Coherence and locality properties; Cache memory organizations, Techniques for reducing cache misses; Virtual memory organization, mapping and management techniques, memory replacement policies. Instruction-level parallelism: Concepts of instruction-level parallelism (ILP), Techniques for increasing ILP; Super scalar, super pipelined and VUW processor architectures; Vector and symbolic processors; Case studies of contemporary microprocessors Multiprocessor Architecture: Taxonomy of parallel architectures; Centralized shared-memory architecture, synchronization, memory consistency, interconnection networks; Distributed shared-memory architecture, Cluster computers. Non von Neumann Architectures: Data flow Computers, Reduction computer architectures, Systolic Architectures.

Suggested Text Books & References

- Hwang, K. "Advanced computer architecture with Parallel programming", McGraw Hill.
- Patterson D.A and Hennessy, J.L. "Computer architecture a quantitative approach", 2nd Edition, Morgan Kaufman, 1996.
- Stone, H.S. "Advanced Compter Architecture" , Adision-Wesley, 1989.
- Siegel, H.J."Interconnection Network for Large Scale Parallel Processing", 2nd Edition, McGraw Hill. 1990.

(CS1605) Operating System II (3-1-0)

System Administration: Understand configuration of H/W, configuration of Kernel, Setting up of serial H/W, configuration of TCP/IP Networking, Name Service & Resolve configuration,

Understanding of various Network Application, Management of NIS, Understanding NFS and AFS, configurations of Mail, configuration NNTP/TIN, file System & Quota Management

Suggested Text Books & References

- * Linux Administration.
- * Hpx Administration Manual
- * DELALPHA Administration Manual.
- * Taninbaum: Modern Operating System

(CS1606) Analysis and Design of Algorithm (3-1-0)

Algorithms and Complexity - asymptotic notations, orders, worst-case and average-case, amortized complexity.

Basic Techniques - divide & conquer, dynamic programming, greedy method, backtracking, branch and bound, randomization. Data Structures - heaps, search trees, union-find problems. Applications - sorting & searching, combinatorial problems, optimization problems, computational geometric problems, string matching. Graph Algorithms - BFS and DFS, connected components, spanning trees, shortest paths, max-flow. NP-completeness.

Approximation algorithms.

Laboratory: Implementation of algorithms covered in class: This will involve running the algorithms under varying Input sets and measuring running times, use of different data structures for the same algorithm (wherever applicable) to see its effect on time and space, comparison of different algorithms for the same problem etc.

Suggested Text Books & References

- Horowitz E. & Sahni, S, "Fundamental of Computer Algorithm", Galgoyia.
- Aho, Hopcroft & Ullman, " The DESIGN & ANALYSIS OF ALGORITHM". Adision-Wesley.
Sedgewick, "Algorithms in C".

VI-SEMESTER PRACTICAL

(0-0-3)

6 th Semester		
SL. NO.	Name of Lab	List of Experiments
CS1607-P	System Administration	<ol style="list-style-type: none">1) TCP/IP configuration2) Configuration of mail server3) Configuration of file server4) Configuration of print server5) Study of Novel NetWare

(0-0-3)

6 th Semester		
SL. NO.	Name of Lab	List of Experiments
CS1608-P	Internet Lab	<ol style="list-style-type: none">1) Configuration of IIS server2) Study of scripting Language3) Static Web Page designing4) Dynamic Web Page designing

(0-0-3)

6 th Semester		
SL. NO.	Name of Lab	List of Experiments
CS1609-P	Computer Graphics	<ol style="list-style-type: none">1) Creation of lines and Plane2) Creation of different polygons3) Implementation of different polygons filling algorithms4) Implementation of different clipping algorithm

(0-0-3)

6 th Semester		
SL. NO.	Name of Lab	List of Experiments
CS1610-P	Language Processor	<ol style="list-style-type: none">1. Write a Compiler for a small language.2. Design a predictive parser for small language.3. Design a Scanner(Lex, Flex)4. Design a Parser.(Yacc/Busy)5. Study of code Optimisation.