Semester VII

COMP 7011: Internet Technology (Theory)

Course Objective:
The core objectives of an Internet Technology course are to provide a comprehensive understanding of the Internet's architecture, TCP/IP protocols, and web development fundamentals (HTML, CSS, JavaScript, JSP). Students learn to build secure, dynamic web applications, analyze network traffic, and understand key concepts like DNS, routing, and HTTP.

Course Learning Outcomes:

After undergoing the course, Students will be able to:

- Implement modular and efficient programs using Object-Oriented Programming (OOP) principles.
- Develop interactive web interfaces by applying JavaScript data types, control structures, and Event Handling techniques to respond to user actions.
- Establish and manage secure communication between Java applications and relational databases using the JDBC API.
- Design and deploy dynamic, data-driven web pages using Java Server Pages (JSP)**.**
- Utilise Implicit JSP Objects, conditional processing, and error-handling mechanisms to manage session data and share information across requests and users.

Credit: 04 **60 Hours**

Syllabus:

Java: Use of Objects, Array and Array List class **(5 Hrs)**

JavaScript: Data types, operators, functions, control structures, events and event handling. **(15 Hrs)**

JDBC: JDBC Fundamentals, Establishing Connectivity and working with connection interface, working with statements, Creating and Executing SQL Statements, Working with Result Set Objects. **(10 Hrs)**

JSP: Introduction to Java Server Pages, HTTP and Servlet Basics, The Problem with Servlets, The Anatomy of a JSP Page, JSP Processing, JSP Application Design with MVC, Setting Up the JSP Environment, Implicit JSP Objects, Conditional Processing, Displaying Values, using an expression to Set an Attribute, Declaring Variables and Methods, Error Handling and Debugging, Sharing Data Between JSP Pages, Requests, and Users, Database Access. **(20 Hrs)**

Java Beans: Java Beans Fundamentals, JAR files, Introspection, developing a simple Bean, Connecting to DB **(10 Hrs)**

Recommended Books:

1. Ivan Bayross, Web Enabled Commercial Application Development Using Html, Dhtml, javascript, Perl Cgi , BPB Publications, 2009.
2. Cay Horstmann, BIG Java, Wiley Publication, 3rd Edition., 2009
3. Herbert Schildt, Java 7, The Complete Reference, , 8th Edition, 2009.
4. Jim Keogh, The Complete Reference J2EE, TMH, , 2002.
5. O'Reilly, Java Server Pages, Hans Bergsten, Third Edition, 2003.

BCA 7011: Internet Technology (Practical)

Credit: 02 **60 Hours**

Create event driven program for following:

1. Print a table of numbers from 5 to 15 and their squares and cubes using alert.
2. Print the largest of three numbers.
3. Find the factorial of a number n.
4. Enter a list of positive numbers terminated by Zero. Find the sum and average of these numbers.
5. A person deposits Rs 1000 in a fixed account yielding 5% interest. Compute the amount in the account at the end of each year for n years.
6. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list

COMP-7012 Compiler Design (Theory)

Credit:4 60 Hour

Course Objectives

The basic objective of the compiler design course is to understand the basic principles of compiler design, its various constituent parts, algorithms, and data structures required to be used in the compiler. It also aims to understand the use of basic compiler-building tools.

Course Learning Outcomes

On successful completion of the course, the students will be able to:

1. Describe the concepts and different phases of compilation.
2. Represent language tokens using regular expressions and context free grammars.
3. Describe the working of lexical analyzers.
4. Understand the working of different types of parsers and parse a particular string.
5. Describe intermediate code representations using syntax trees and DAG's as well as use this knowledge to generate intermediate code in the form of three address code representations.
6. Apply optimization techniques to intermediate code and generate machine code for high level language program.
7. Use Lex and Yacc automated compiler generation tools.

Syllabus

Unit 1 Introduction: Overview of compilation, Phases of a Compiler. (5 Hrs)

Unit 2 **Lexical Analysis:** Role of a Lexical analyzer, Specification and recognition of tokens, Symbol table, Error reporting, Regular expressions and definitions, Lexical Analyzer Generator-Lex.(15 Hrs)

Unit 3 **Syntax Analysis:** CFGs, left recursion, left factoring, Top-down parsing- LL parser, Bottom-up parsing- LR parser, Parser Generator-yacc.(15 Hrs)

Unit 4 Intermediate representations: Syntax Directed Definitions, Evaluation Orders for Syntax Directed Definitions, Intermediate Languages: Syntax Tree, Three Address Code, Types and Declarations, Translation of Expressions, loops and conditional statements, Type Checking. (15 Hrs)

Unit S **Storage organization & Code generation:** Activation records, stack allocation, Issues in Code Generation — Design of a simple Code Generator. (15 Hrs)

Unit 6 **Code optimization:** Principal sources of optimization, Peephole optimization. (10 Hrs)

Reference Books

1. Aho, A., Lam, M., Sethi, R., & Ullman, J. D. *Compilers: Principles, Techniques, and Tools,* 2ND edition, Addison Wesley, 2006.
2. V Raghvan, *Principles of Compiler Design,* TMH, 2010.
3. Santanu Chattopadhayay, *Compiler Design,* PHI, 2005.

Compiler Design (Practical)

Credit:2 ***60 Hour***

Suggested Practical List

Write a Lex program to count the number of lines and characters in the input file.

1. Write a Lex program to count the number of vowels and consonants in a given string
2. Write a Lex program that implements the Caesar cipher: it replaces every letter with the one three letters after in alphabetical order, wrapping around at Z. e.g. a is replaced by d, b by e, and so on z by c.
3. Write a Lex program that finds the longest word (defined as a contiguous string of upper and lower case letters) in the input.
4. Write a Lex program that distinguishes keywords, integers, floats, identifiers, operators, and comments in any simple programming language.
5. Write a Lex program to count the number of words, characters, blank spaces and lines in a C file.
6. Write a Lex specification program that generates a C program which takes a string “abcd” and prints the following output

   abcd

   abc
7. Write a Lex program to recognize a valid arithmetic expression.

COMP 7013 Advanced Data Structure & Algorithms (Theory)

Credit:4 **60 Hours**

Course Objectives

(i) Gain in-depth knowledge of complex data organization techniques like AVL trees, B-trees, hash tables, and graphs to efficiently store and retrieve data.

(ii) Learn mathematical and analytical tools, such as asymptotic notations (Big Oh, Omega, Theta), to determine the time and space efficiency of algorithms and choose the best method for a given problem.

(iii) Become proficient in various algorithm design paradigms, including divide and conquer, greedy methods, dynamic programming, backtracking, and branch and bound, to solve complex computational challenges.

(iv) Strengthen critical thinking and problem-solving abilities by breaking down large, complex problems into manageable steps and devising efficient, optimized solutions.

(v) Develop programming skills to implement learned data structures and algorithms effectively, ensuring they perform well in real-world applications

Course Outcomes:

After undergoing the course, Students will be able to understand:

- Students should develop a sound theoretical understanding of advanced algorithms and practical problem-solving skills using them.
- Students should develop basic knowledge of a wide range of advanced algorithm design techniques including dynamic programming, linear programming, approximation algorithms, and randomized algorithms.
- Students should develop basic advanced algorithm analysis skills for analysing the approximation ratio of approximation algorithms and the probability of randomized algorithms.
- Students should gain a good understanding on a wide range of advanced algorithmic problems, their relations and variants, and application to real-world problems.

Syllabus

UNIT-I: Introduction to Advanced Data Structures: Introduction to Data Structures, Linear Data Structures, Binary Trees, insertion and deletion in BST. Query in BST, Thread Binary Trees, Balancing Trees, AVL Trees, Red black Trees, B Tree, Splay Trees, Heap, different data structures for heap, double ended heap, Priority Queue, Data Structure for Disjoint-Sets. K-D Tree, removal of recursion. Introduction to Object Oriented Data Structures. [12 Hrs]

UNIT-II: Analysis of Algorithm: Solving problem using computer, general problem solving approach – STAIR, reduction to known problem, meaning of algorithm, steps in development of algorithm, evolution of algorithms, design by analysis, design by synthesis, algorithm design patterns and frameworks,

Correctness of an algorithm; concept of space and time complexity; asymptotic analysis, amortization, randomization and experimental analysis; best, worst and average case, order of growth, asymptotic notations, upper and lower bounds; analysing rate of growth of a function with that of known functions, comparing algorithms w. r. t. order of growth; applying algorithm analysis concepts for complexity analysis of different sorting techniques. [6 Hrs]

UNIT-III: Divide and Conquer: Introduction to top – down approach, divide – and – conquer approach, identifying problems where divide – and – conquer suits / is unsuitable, analysis of divide and conquer algorithms, recurrences, the master method for solving recurrences. [6Hrs]

UNIT-IV: Dynamic Programming : Principle of optimality, bottom – up approach to problem solving, comparison with top – down approach, application of dynamic programming to solve problems using optimal sequence of decisions, comparison with brute – force method for finding optimal solution, possibility of multiple sequences in dynamic programming, overlapping sub – problems and optimal substructures, avoiding further exploration of sub – optimal decisions using memorizations, examples of problems; correctness of dynamic programming; analysis of dynamic programming. [12 Hrs]

UNIT-V: Greedy Algorithms: Another approach to find optimal solution by generating only one decision sequence (locally best choice when all available choices are exhausted), optimal substructure and greedy choice; greedy algorithms vs. dynamic programming; examples of problems; analysis of greedy algorithm; matroids; correctness of the greedy algorithm on matroids. [6 Hrs]

UNIT-VI: Sorting and Searching Techniques, Elementary sorting techniques– Merge Sort, Advanced Sorting techniques - Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Searching Techniques, Medians & Order Statistics, complexity analysis, lower bound on sorting. [6 Hrs]

UNIT-VII: Graph algorithms: Introduction to graphs and their representations, Graph traversal, DFS, BFS and applications, Connectivity, strong connectivity, bi-connectivity, Minimum spanning tree, Shortest paths, all pair shortest path, Matching, Network flow. [6Hrs]

UNIT-VIII: String Algorithms: Naive string matching, Rabin-Karp, Matching with FA, KMP algorithm, Boyer-Moore algorithm, LCS, String Compression Algorithm, Sequence alignment, edit distance, Levenstein distance, Needleman–Wunsch algorithm, gap, gap penalty. [6Hrs]

REFERENCE BOOKS:

1.*Introduction to Algorithms – A Creative Approach*: Udi Manber, Addison – Wesley Professional

2.*Introduction to Algorithms:* Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, MIT

3.*Algorithms:* Richard Johnsonbaugh, Marcus Schaefer, Pearson

4.*Fundamentals of Computer Algorithms:* Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, University Press

5.*Design and Analysis of Algorithms:* Parag Himanshu Dave, Himanshu Bhalchandra Dave, Pearson

6.*Analysis of Algorithms: Computational Methods & Mathematical Tools:* Micha Hofri, Oxford University Press, USA

Advanced Data Structure & Algorithms (Practical)

Credit: 2 **60 Hrs**

All programs should be developed in C/Java/Python

1. Implement Quick Sort (The program should report the number of comparisons)
2. Implement Merge Sort(The program should report the number of comparisons)
2. Implement Heap Sort(The program should report the number of comparisons)
4. Implement Radix Sort (The program should report the number of comparisons)
5. Implement Counting sort (The program should report the number of comparisons)
6. Implement Threaded Binary Tree, AVL tree , Red Black Tree, B tree, Splay Trees, Heap and perform different operation on them.
7. Implement Naive string matching, Rabin-Karp, Matching with FA, KMP algorithm, Boyer-Moore algorithm, LCS, String Compression Algorithm, Sequence alignment, edit distance, Levenstein distance, Needleman–Wunsch algorithm.
8. Implement Breadth-First Search in a graph.
9. Implement Depth-First Search in a graph.
10. Write a program to determine the minimum spanning tree and shortest path of a graph

COMP 7014 Soft Computing (Theory)

Credit: 4 **60 Hrs**

Syllabus

Module-I (10 Hrs)
Introduction: Introduction to soft computing; introduction to fuzzy sets, rough setsm and fuzzy logic systems; introduction to biological and artificial neural network; introduction to Genetic Algorithm.

Module-II: Fuzzy Logic (20 Hrs)
Classical Sets and Fuzzy Sets and Fuzzy relations : Operations on Classical sets, properties of classical sets, Fuzzy set operations, properties of fuzzy sets, cardinality, operations, and properties of fuzzy relations.
Membership functions : Features of membership functions, standard forms and boundaries, different fuzzification methods.
Fuzzy to Crisp conversions: Lambda Cuts for fuzzy sets, fuzzy Relations, Defuzzification methods.

Classical Logic and Fuzzy Logic: Classical predicate logic, Fuzzy Logic, Approximate reasoning and Fuzzy Implication

Fuzzy Rule based Systems: Linguistic Hedges, Fuzzy Rule based system – Aggregation of fuzzy Rules, Fuzzy Inference System, Mamdani Fuzzy Models – Takagi-Sugeno Fuzzy Models.

Module-III: Neural Networks (20 Hrs)

Introduction to Neural Networks: Advent of Modern Neuroscience, Classical AI and Neural Networks, Biological Neurons and Artificial neural network; model of artificial neuron.

Learning Methods : Hebbian, competitive, Boltzman.

Neural Network models: Perceptron, Adaline and Madaline networks; single layer network; Delta rule and back-propagation and multi layer networks.

Competitive learning networks: Kohonen self organizing networks (SOM), Hebbian learning; Hopfield Networks, Bidirectional Associative Memory (BAM).

Module-IV: Genetic Algorithms (10 Hrs)

Genetic Algorithms: Simple GA, crossover and mutation, Multi-objective Genetic Algorithm (MOGA).

Text Books:

1. Fuzzy logic with engineering applications, Timothy J. Ross, John Wiley and Sons. 2. S. Rajasekaran and G.A.V.Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithms", PHI
3. Principles of Soft Computing , S N Sivanandam, S. Sumathi, John Wiley & Sons
4. Genetic Algorithms in search, Optimization & Machine Learning by David E. Goldberg
5. Neuro-Fuzzy and Soft computing, Jang, Sun, Mizutani, PHI
6. Neural Networks: A Classroom Approach,1/e by Kumar Satish, TMH,
7. Genetic Algorithms in search, Optimization & Machine Learning by David E. Goldberg, Pearson/PHI
8. A beginners approach to Soft Computing, Samir Roy & Udit Chakraborty,
9. Soft Computing: Fundamentals and Applications, by D. K. Pratihar.

Soft Computing Lab (Practical)

Credit: 2 **60 Hrs**

All programs should be implemented using C / Java / Python

Practical exercises based on concepts listed in theory, such as:

1. Applying an ANN to train a model

2 Apply a Kohonen SOM

3. Apply a Hopfield network to memorize and rectify patterns

4 Apply BAM to memorize

5. Apply Genetic Algorithms to solve an optimization problem

6. Applying ANN for pattern recognition tasks

Minor Paper
COMP 7021: Database Management System (Theory)

Course Objective:

A Database Management System (DBMS) course aims to provide a comprehensive understanding of how to design, implement, and manage databases effectively. Core objectives include mastering relational models, constructing SQL queries, applying normalization to reduce redundancy, and managing transactions and concurrency control for data integrity.

Course Learning Outcomes:

After undergoing the course, Students will be able to:

- Differentiate between traditional file systems and the database approach, explaining how the three-schema architecture provides data independence.
- Construct Entity-Relationship (ER) models to capture complex business requirements, defining entity sets, attributes, and relationship constraints.
- Execute complex data retrieval and manipulation tasks using Relational Algebra and industry-standard SQL (Structured Query Language).
- Transform ER models into relational schemas and apply Normalization (up to BCNF) to eliminate redundancy and ensure data integrity through functional dependency analysis.

Credit: 03 **45 Hours**

1.Introduction (6 Hrs)

Characteristics of database approach, data models, database system architecture and data independence.

2.Entity Relationship (ER) Modeling (8 Hrs)

Entity types, relationships, constraints.

3.Relation data model (20 Hrs)

Relational model concepts, relational constraints, relational algebra, SQL queries

4. Database design (15 Hrs)

Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition,

Normal forms (upto BCNF).

5. **Transaction Processing** (3 Hrs)
   ACID properties, concurrency control & recovery

6. **File Structure and Indexing** (8 Hrs)
   Operations on files, File of Unordered and ordered records, overview of File organizations, Indexing structures for files(Primary index, secondary index, clustering index), Multilevel indexing using B and B+ trees.

References Book :

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.
2. R. Ramakrishanan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002.
3. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts 6th Edition, McGrawHill, 2010.
4. R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education, 2013.

COMP 7021: Database Management System (Practical)

Credit: 01 30 Hours

Create and use the following database schema to answer the given queries.

EMPLOYEE Schema

Field	Type	NULL	KEY	DEFAULT
Eno	Char(3)	NO	PRI	NIL
Ename	Varchar(50)	NO		NIL
Job_type	Varchar(50)	NO		NIL
Manager	Char(3)	Yes	FK	NIL
Hire_date	Date	NO		NIL
Dno	Integer	YES	FK	NIL
Commission	Decimal(10,2)	YES		NIL
Salary	Decimal(7,2)	NO		NIL

DEPARTMENT Schema

Field	Type	NULL	KEY	DEFAULT
Dno	Integer	No	PRI	NULL
Dname	Varchar(50)	Yes		NULL

Location Varchar(50) Yes New Delhi

Query List

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.
4. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.
5. Query to display the Employee Name and Salary of all the employees earning more than $2850.
6. Query to display Employee Name and Department Number for the Employee No= 7900.
7. Query to display Employee Name and Salary for all employees whose salary is not in the range of $1500 and $2850.
8. Query to display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.
9. Query to display Name and Hire Date of every Employee who was hired in 1981.
10. Query to display Name and Job of all employees who don,,t have a current Manager.
11. Query to display the Name, Salary and Commission for all the employees who earn commission.
12. Sort the data in descending order of Salary and Commission.
13. Query to display Name of all the employees where the third letter of their name is _A,,.
14. Query to display Name of all employees either have two _R,,s or have two _A,,s in their name and are either in Dept No = 30 or their Manger,,s Employee No = 7788.

15. Query to display Name, Salary and Commission for all employees whose Commission Amount is 14 greater than their Salary increased by 5%.
16. Query to display the Current Date.
17. Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.
18. Query to display Name and calculate the number of months between today and the date each employee was hired.
19. Query to display the following for each employee <E-Name> earns < Salary> monthly but wants < 3 * Current Salary >. Label the Column as Dream Salary.
20. Query to display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with _J,,, ,,A,, and _M,,.
21. Query to display Name, Hire Date and Day of the week on which the employee started.
22. Query to display Name, Department Name and Department No for all the employees.
23. Query to display Unique Listing of all Jobs that are in Department # 30.
24. Query to display Name, Dept Name of all employees who have an _A,, in their name.
25. Query to display Name, Job, Department No. And Department Name for all the employees working at the Dallas location.
26. Query to display Name and Employee no. Along with their Mangers Name and the Managers employee no; along with the Employees,, Name who do not have a Manager.
27. Query to display Name, Dept No. And Salary of any employee whose department No. and salary matches both the department no. And the salary of any employee who earns

a commission.

28. Query to display Name and Salaries represented by asterisks, where each asterisk (*) signifies $100.
29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees
30. Query to display the number of employees performing the same Job type functions.
31. Query to display the no. of managers without listing their names.
32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.
33. Query to display Name and Hire Date for all employees in the same dept. as Blake.
34. Query to display the Employee No. And Name for all employees who earn more than the average salary.
35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a “T”
36. Query to display the names and salaries of all employees who report to King.
37. Query to display the department no, name and job for all employees in the Sales department