COMPILER DESIGN

Subject Code- BCSES1-501 **L T P C** **Duration – 60 Hrs.**

3 1 0 4

COURSE OBJECTIVE:

This course will help students to understand the process involved in a compiler. This course will make student aware about the working of top down and bottom up parsers. This will help students to better understand the different phases of compilation and generation of target code for a machine.

COURSE OUTCOMES:

1. For a given grammar specification, develop the lexical analyser.
2. For a given parser specification design top-down and bottom-up parsers.
3. Use syntax directed translation schemes to develop intermediate code.
4. Learn algorithms to generate code for a target machine

COURSE CONTENTS:

UNIT I (10 Hrs)

Introduction: Phases of compilation and overview.

Lexical Analysis (scanner): Regular languages, finite automata, regular expressions, from regular expressions to finite automata, scanner generator (LEX).

UNIT II (20 Hrs)

Syntax Analysis (Parser): Context-free languages and grammars, push-down automata, LL(1) grammars and top-down parsing, operator grammars, LR(O), SLR(1), LR(1), LALR(1) grammars and bottom-up parsing, ambiguity and LR parsing, LALR(1) parser generator (YACC)

Semantic Analysis: Attribute grammars, syntax directed definition, evaluation and flow of attribute in a syntax tree. Symbol Table: Its structure, symbol attributes and management. Run-time environment: Procedure activation, parameter passing, value return, memory allocation, and scope.

UNIT III (15 Hrs)

Intermediate Code Generation: Translation of different language features, different types of intermediate forms.

Code Improvement (optimization): control-flow, data-flow dependence etc.; Code improvement local optimization, global optimization, loop optimization, peep-hole optimization etc.

UNIT IV (15 Hrs)

Target code generation: Architecture dependent code improvement: instruction scheduling, Introduction to code generation, Target Machine, Register allocation, issues in code generation, A simple code generation algorithm.

RECOMMENDED BOOKS

1. V. Aho, R. Sethi, and J. Softec, D. Ullman, 'Compilers: Principles, Techniques and Tools', 2nd Edn., Addison-Wesley, **2006**.
2. Fischer and R. LeBlanc, 'Crafting a Compiler', Benjamin Cummings, **2009.**
3. C. Fischer and R. LeBlanc, 'Crafting a Compiler in C', Benjamin Cummings, **1991**.
4. C. Holub, 'Compiler Design in C', Prentice-Hall Inc., **1993**.

5. 'Modern Compiler Implementation in C: Basic Design', Cambridge Press, **2004**.
6. 'Modern Compiler Implementation in Java: Basic Design', 2nd Edn., Cambridge Press, **2002**.
7. Fraser and Hanson. A Retargetable C, 'Compiler: Design and Implementation', Addison-Wesley, **1995**.

DATABASE MANAGEMENT SYSTEM

Subject Code- BCSES1- 502 **L T P C** **3 0 0 3** **Duration – 45 Hrs.**

COURSE OBJECTIVE:

This course will help student to understand the concepts used in database management systems. They will also help to create database using DDL and DML. They will learn to implement database security and various advanced topics will also be covered.

COURSE OUTCOMES:

1. To be able to learn different DBMS languages and data models.
2. For a given specification construct the SQL queries for Open source and Commercial DBMS -MYSQL, ORACLE, and DB2.
3. For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.
4. Implement database security.

COURSE CONTENTS:

UNIT I (11 Hrs)

Database system architecture: introduction, Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).
Data models: Entity-relationship model, network model, relational and object oriented data models, integrity constraints.

UNIT II (11 Hrs)

Relational query languages: Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, introduction to MYSQL, ORACLE, DB2, SQL server.
Relational database design: Domain and data dependency, Normal forms, Dependency preservation, Lossless design.

UNIT III (12 Hrs)

Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.
Transaction processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic
Concurrency Control schemes,

UNIT IV (11 Hrs)

Database recovery: Introduction, log based recovery, shadow page recovery.
Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, introduction to SQL injection.
Advanced topics: Introduction to Object oriented, Distributed databases.

RECOMMENDED BOOKS

1. "Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F.Korth, S. Sudarshan, McGraw-Hill.

2. "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
3. "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education
4. "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

FORMAL LANGUAGE AND AUTOMATA THEORY

Subject Code- BCSES1-503 **L T P C** **Duration – 45 Hrs.**
3 0 0 3

COURSE OBJECTIVE:
1. Develop a formal notation for strings, languages and machines.
2. Design finite automata to accept a set of strings of a language.
3. Identify the hierarchy of formal languages, grammars and machines.

COURSE OUTCOMES:
1. Design finite automata to accept a set of strings of a language.
2. Design context free grammars to generate strings of context free language.
3. Design Turing machine for accepting context sensitive languages.
4. To learn Rice's theorem.

COURSE CONTENTS:

UNIT I (11 Hrs)

Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages.

Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata.

UNIT II (12 Hrs)

Context-free languages and pushdown automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs.

UNIT III (12 Hrs)

Context sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.

Turing machines: The basic model for Turing machines (TM),Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators.

UNIT IV (10 Hrs)

Undecidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice's theorem, undecidable problems about languages.

RECOMMENDED BOOKS

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia.
2. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.
3. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.
4. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.
5. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill.

DESIGN &ANALYSIS OF ALGORITHMS

Subject Code- BCSES1-504 **L T P C** **Duration – 60 Hrs.**

3 1 0 4

COURSE OBJECTIVE:

1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Demonstrate a familiarity with major algorithms and data structures.
4. Apply important algorithmic design paradigms and methods of analysis.
5. Synthesize efficient algorithms in common engineering design situations.

COURSE OUTCOMES:

1. For a given algorithms analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms.
2. Describe the greedy paradigm and explain when an algorithmic design situation calls for it.
3. Describe the different graph and tree traversal algorithms.
4. Describe the computability of problem using Cook's theorem.

COURSE CONTENTS:

UNIT I (15 Hrs)

Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behaviour; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem.

UNIT II (15 Hrs)

Fundamental Algorithmic Strategies: Brute-Force, Greedy, Dynamic Programming, Branch and Bound and Backtracking methodologies for the design of algorithms; Illustrations of these techniques for Problem Solving, Bin Packing, Knap Sack TSP. Heuristics –characteristics and their application domains.

UNIT III (15 Hrs)

Graph and Tree Algorithms: Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm.

UNIT IV (15 Hrs)

Tractable and Intractable Problems: Computability of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Cook's theorem, Standard NP-complete problems, and Reduction techniques. Introduction to recent advancements in design and analysis of algorithms.

RECOMMENDED BOOKS:

1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2. Fundamentals of Algorithms – E. Horowitz et al.
3. Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson.
4. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Michael T Goodrich and Roberto Tamassia, Wiley.
5. Algorithms—A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA.

DATABASE MANAGEMENT SYSTEM LABORATORY

Subject Code- BCSES1-505

L	T	P	C
0	0	4	2

COURSE OBJECTIVE:

To learn the implementation of SQL queries to perform DBMS operations.

COURSE OUTCOMES:

1. To understand basic DDL, DML, DCL commands
2. To understand the SQL queries using SQL operators
3. To understand the concept of relational algebra, date and group functions
4. To implement checkpoints.

PRACTICALS:

1. Write the queries for Data Definition Language (DDL) in RDBMS.
2. Write the queries for Data Manipulation Language (DML) in RDBMS.
3. Write the queries for Data Control Language (DCL) in RDBMS.
4. Write SQL queries using logical operators
5. Write SQL queries using SQL operators
6. Write SQL query using character, number, date and group functions
7. Write SQL queries for relational algebra
8. Write SQL queries for extracting data from more than one table
9. Write SQL queries for sub queries, nested queries
10. Concepts for ROLL BACK, COMMIT & CHECK POINTS
11. Case studies on normalization

DESIGN &ANALYSIS OF ALGORITHMS LABORATORY

Subject Code- BCSES1-506 **L T P C**
0 0 2 1

COURSE OBJECTIVE:

1. To get a first-hand experience of implementing well-known algorithms in a high-level language.
2. To be able to compare the practical performance of different algorithms for the same problem.

COURSE OUTCOMES:

1. To perform different operations on integers.
2. To sort number of elements of an array using different sorting techniques.
3. To implement dynamic programming for various problems.
4. To compute convex hull.

PRACTICALS

1. Code and analyse to compute the greatest common divisor (GCD) of two numbers.
2. Code and analyse to find the median element in an array of integers.
3. Code and analyse to find the majority element in an array of integers.
4. Code and analyse to sort an array of integers using Heap sort.
5. Code and analyse to sort an array of integers using Merge sort.
6. Code and analyse to sort an array of integers using Quick sort.
7. Code and analyse Knapsack problem using dynamic programming
8. Code and analyse to find the shortest path for single source shortest path using dynamic programming.
9. Code and analyse to find the shortest path for All pair shortest path using dynamic programming.
10. Code and analyse to do a depth-first search (DFS) on an undirected graph. Implementing an application of DFS such as to find the topological sort of a directed acyclic graph.
11. Code and analyse to do a breadth-first search (BFS) on an undirected graph. Implementing an application of BFS such as (i) to find connected components of an undirected graph, OR (ii) to check whether a given graph is bipartite.
12. Code and analyse to find the minimum spanning tree in a weighted, undirected graph.
13. Code and analyse to find all occurrences of a pattern P in a given string S using KMP Method
14. Code and analyse to compute the convex hull of a set of points in the plane.

COMPUTER GRAPHICS

Subject Code- BCSED1-511 **L T P C** **Duration – 45 Hrs.**
3 0 0 3

COURSE OBJECTIVE:

1. Understanding the fundamental graphical operations and the implementation on computer,
2. To get a glimpse of recent advances in computer graphics.
3. Understanding user interface issues that make the computer easy for the novice to use.

COURSE OUTCOME:

1. Able to learn about the basics of graphics, its applications, uses and Knowledge to draw different shapes in graphics on computer.
2. Ability to apply different 2-D and 3-D transformations on an object.
3. Learn clipping operations and various object filling techniques, different projections techniques. Various hidden surface removal.
4. Knowledge of Rendering techniques, Fractals and different colour models.

COURSE CONTENTS:

UNIT I (12 Hrs)

Introduction: Computer Graphics and its applications, Elements of a Graphics, Graphics Systems: Video Display Devices, Raster Scan Systems, Random Scan Systems, Input devices. **Basic Raster Graphics**: Scan conversion- Point plot technique, Line drawing, Circle generating and Ellipse generating algorithms.

UNIT II (11 Hrs)

Two-dimensional Geometric Transformations: Basic Transformations-Translation, Rotation and Scaling, Matrix Representation and Homogeneous Coordinates, Composite Transformations, Reflection and Shearing transformations.

Elementary 3D Graphics: Matrix Representation of 3D transformations, Plane projections and its types, Vanishing points, Specification of a 3D view.

UNIT III (11 Hrs)

Clipping: Window to viewport transformation, Clipping Operations- Point Clipping, Line Clipping, Polygon Clipping and Text Clipping.

Filling Techniques: Scan line algorithms, Boundary-fill algorithm, Flood-fill algorithm.

Visibility: Image and object precision, Hidden edge/surface removal or visible edge/surface determination techniques; z buffer algorithms, Depth sort algorithm, Scan line algorithm and Floating horizon technique.

UNIT IV (11 Hrs)

Color Models: Properties of Light, Intuitive Color Concepts, RGB Color Model, CMY Color Model, HLS and HSV Color Models, Conversion between RGB and CMY color Models, Conversion between HSV and RGB color models, Color Selection and Applications.

Advance Topics: Introduction of Rendering, Fractals, Gourard and Phong shading.

RECOMMENDED BOOKS:

1. Donald Hearn and M. Pauline Baker, **'**Computer Graphics', 4th Edn.**,** PHI/Pearson Education, **2010**.

2. Zhigand Xiang, Roy Plastock, Schaum's Outlines, **'**Computer Graphics**',** 2nd Edn., Tata Mc-Graw Hill, **2001**.

3. C. Foley, Van Dam, Feiner and Hughes, **'**Computer Graphics Principles & Practice**'**, 3rd Edn., Pearson Education**, 2013.**

4. Roy A. Plastock, Gordon Kalley, 'Computer Graphics', 1st Edn., Schaum's Outline Series, **1986**.

GRAPH THEORY

Subject Code- BCSED1-512 **L T P C** **Duration – 45 Hrs.**
3 0 0 3

COURSE OBJECTIVE:

1. The course is aimed at developing the fundamental Mathematical skills of Graduates that are instrumental for effective understanding of advanced Engineering subjects.
2. The topics introduced will serve as the tools for specialized studies in many Engineering fields, significantly in Field theory, computer technology and Communication Engineering.

COURSE OUTCOMES:

1.To have knowledge of the basic concepts of graph
2.To have a knowledge of classes of graphs and its properties.
3.To have knowledge of graph algorithms.
4.Be exposed to constrained and unconstrained optimization techniques

COURSE CONTENTS:

UNIT I (11 Hrs)

BASICS OF GRAPH THEORY: Graphs - Data structures for graphs – Sub graphs - Operations on Graphs Connectivity - Networks and the maximum flow - Minimum cut theorem.

Trees - Spanning trees - Rooted trees - Matrix representation of graphs.

UNIT II (11 Hrs)

CLASSES OF GRAPHS: Eulerian graphs and Hamiltonian graphs - Standard theorems - Planar graphs - Euler's formula. Five colour theorem - Coloring of graphs - Chromatic number (vertex and edge) properties and examples - Directed graphs.

UNIT III (12 Hrs)

GRAPH ALGORITHM: Computer Representation of graphs - Basic graph algorithms - Minimal spanning tree algorithm - Kruskal and Prim's algorithm. Shortest path algorithms - Dijsktra's algorithm, DFS and BFS algorithms.

UNIT IV (11 Hrs)

OPTIMIZATION TECHNIQUES: Single variable and multivariable optimization - Lagrangian method - Kuhn-Tucker conditions. Random pattern and Random search methods.

RECOMMENDED BOOKS

1. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science.
2. Rao S.S, Engineering Optimization: Theory and Practice, New AgeInternational Pvt. Ltd., 3rd Edition1998.

WEB TECHNOLOGIES

Subject Code- BCSED1-513 **L T P C** **Duration – 45 Hrs.**
3 0 0 3

COURSE OBJECTIVE:

1. Designing the HTML pages along with style sheets

2. Familiar with client and server side scripting.
3. Able to develop a web application.
4. Students will gain the skills and project-based experience needed for entry into web application and development careers.

COURSE OUTCOMES:

1. To understand the HTML and Style Sheets
2. To have knowledge of client side scripting using JSP.
3. To understand the basics and object oriented concepts of PHP.
4. To access database using PHP programming.

COURSE CONTENTS:

UNIT – I (12 Hrs)

Introduction, History of HTML, Structure of HTML Document: Text Basics, Structure of HTML Document: Images and Multimedia, Links and webs, Document Layout, Creating Forms, Frames and Tables.

Style sheets : Need for CSS, introduction to CSS, basic syntax and structure, using CSS, background images, colors and properties, manipulating texts, using fonts, borders and boxes, margins, padding lists, positioning using CSS.

UNIT – II (09 Hours)

Javascript : What is Javascript, Client side scripting, Data types, variables, operators, conditional statements, loops and repetition, array object, date object, string object, Document object model - Event handling.

UNIT – III (12 Hours)

Introduction to PHP, Writing PHP, Control Structures, if-else, switch, ?operator, while, do-while, for, for each, break, continue, goto, exit, arrays, functions

Introduction – Declaring a class – Objects – constructor – Destructor – Public ,private, protected – Static properties and method – Inheritance

UNIT – IV (12 Hours)

Working with data, form element, Get, Post, Request, Cookies, Sessions and Access Control: Cookies - PHP and HTTP Authentication – sessions - using Auth_HTTP to Authenticate.

Working MySQL with PHP-database connectivity- usage of MYSQL commands in PHP, processing result sets of queries- handling errors-debugging and diagnostic functions-validating user input through Database layer and Application layer- formatting query output.

RECOMMENDED BOOKS:

1. PHP: The Complete Reference, “Steven Holzner”, Tata McGraw Hill.
2. Programming PHP, “Kevin Tetroi”, O' Reilly.
3. Robin Nixon, Learning PHP, MySQL, and JavaScript, Shroff/O'Reilly
4. VikramVaswani, “PHP and MySQL”, Tata McGraw-Hill, 2005
5. Marty Hall, Larry Brown, ‘Core Servlets and Java Server Pages Vol. 1: Core Technologies’, 2nd Edn., Pearson, 2003.
6. Dietel, Niet, ‘Internet and World Wide Web – How to Program’, 5th Edn., PHI/Pearson Education, 2011.

7. Wang, 'An Introduction to web Design and Programming', 1st Edn.,Cengage COURSE, 2003.
8. Thomas A Powell, The Complete Reference HTML & CSS, 5th Edition, Tata McGraw Hill
9. Laura Lemay, Rafe Colburn, Jennifer Kyrnin, 'Mastering HTML, CSS &Javascript Web Publishing', Sams Teach Yourself.
10. Sebesta, 'Programming World Wide Web', 4th Edn., Pearson, 2008

JAVA PROGRAMMING

Subject Code- BCSED1-514 **L T P C** **Duration – 45 Hrs.**
3 0 0 3

COURSE OBJECTIVE:
1. To learn the basic and advanced concepts of Java Programming language.
2. To experience the working environment required for programming in Java language and enhances their programming skills.

COURSE OUTCOMES:
1. To learn the basics of Java and to understand the implementation of Classes and Inheritance with respect to Java.
2. To describe the concept of handling of exceptions and multithreading.
3. To understand how to implement I/O, Applets and Graphics in Java
4. To comprehend the advanced topics of Java Programming

COURSE CONTENTS:

UNIT-I (12 Hrs)

Introduction to Java: Features of Java, difference between Java and C++, JVM, Bytecode, data types, variables, arrays, Type Conversion and Casting.

Classes and Inheritance: Class Fundamentals, methods, constructors, garbage collection, this keyword, Overloading constructors, Nested and Inner classes. Basics and types of inheritance, Method Overriding, Abstract Classes, final keyword, packages and interfaces.

UNIT-II (12 Hrs)

Exception Handling: Basics, Exception Types, uncaught exceptions, try and catch, throwing exceptions.

Introduction to Multithreading: Java thread model, thread priorities, synchronization, interthread communication, creating, suspending, resuming threads.

UNIT-III (12 Hrs)

I/O: Input/Output, reading and writing files.

Applets and Graphics: Applet basics, Applet class, Applet initialization and termination, event handling, keyboard and mouse events, AWT class, Layout managers, panels, canvases, Frame windows, drawing lines, rectangles, ellipses.

UNIT-IV (09 Hrs)

Advance Concepts: JDBC Connectivity, Introduction to Java Beans, Java Swings, Java Server Pages.

RECOMMENDED BOOKS:

1. Patrick Naughton & Herbert Schildt, 'The Complete Reference Java 2', 5th Edn., Tata McGraw Hill, **2002**.
2. Balagurusamy, 'Programming in JAVA', BPB Publications, **2006**.
3. Deitel and Deitel, 'Java: How to Program', 10th Edn., Pearson Education, **2014**