CO4	3	3	2			1	2		3	2	
CO5	2										

Compiler Design

(Please fill the above with Levels of Correlation, viz., L-1, M-2, H-3)

SYLLABUS:

UNIT-1: Lexical Analysis: Language Processors, Structure of a Compiler, Lexical Analysis, The Role of the Lexical Analyzer, Bootstrapping, Input Buffering, Specification of Tokens, Recognition of Tokens, Lexical Analyzer Generator-LEX, Finite Automata, Regular Expressions and Finite Automata, Design of a Lexical Analyzer Generator.

UNIT-2: Syntax Analysis: The Role of the Parser, Context-Free Grammars, Derivations, Parse Trees, Ambiguity, Left Recursion, Left Factoring, **Top Down Parsing:** Pre Processing Steps of Top Down Parsing, Backtracking, Recursive Descent Parsing, LL (1) Grammars, Non-recursive Predictive Parsing, Error Recovery in Predictive Parsing.

UNIT-3: Bottom Up Parsing: Introduction, Difference between LR and LL Parsers, Types of LR Parsers, Shift Reduce Parsing, SLR Parsers, Construction of SLR Parsing Tables, More Powerful LR Parses, Construction of CLR (1) and LALR Parsing Tables, Dangling Else Ambiguity, Error Recovery in LR Parsing, Handling Ambiguity Grammar with LR Parsers.

UNIT-4: Syntax Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Applications of Syntax Directed Translation, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD's. **Intermediate Code Generation:** Variants of Syntax Trees, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Backpatching, Intermediate Code for Procedures.

UNIT-5: Run Time Environments: Storage Organization, Run Time Storage Allocation, Activation Records, Procedure Calls, Displays, **Code Optimization:** The Principle Sources of Optimization, Basic Blocks, Optimization of Basic Blocks, Structure Preserving Transformations, Flow Graphs, Loop Optimization, Data-Flow Analysis, Peephole Optimization, **Code Generation:** Issues in the Design of a Code Generator, Object Code

Design & Analysis of Algorithms

SYLLABUS:

UNIT I: Introduction: Algorithm- introduction, Algorithm characteristics, Pseudo code

---

R20
Curriculum Structure

B. Tech
Computer Science & Engineering

Department of Computer Science & Engineering
University College of Engineering, JNT University Kakinada

Conventions Recursive Algorithm, Performance Analysis, Space Complexity, Time Complexity, Probabilistic Complexity, Asymptotic Notation, Practical Complexities, Big and little- O notations.

UNIT II: Divide and Conquer: General Method, Defective chessboard, Binary Search, finding the maximum and minimum, Merge sort, Quick sort. **The Greedy Method:** The general Method, knapsack problem, Job sequencing with deadlines, minimum-cost spanning Trees, Optimal Merge Patterns, Single Source Shortest Paths.

UNIT III: Dynamic Programming: General Method, All pairs-shortest paths, String Editing, 0/1 knapsack, Reliability Design, Floyd-Warshall algorithm for shortest path.

UNIT IV: Backtracking: The General Method, The 8-Queens problem, sum of subsets, Graph coloring, Hamiltonian cycles, knapsack problem.

Branch and Bound: The Method, Least cost (LC) Search, The 15- Puzzle: an Example, Control Abstraction for LC-Search.

UNIT V: Bounding, FIFO Branchand-Bound, LC Branch and Bound, 0/1 Knapsack Problem, Traveling Salesperson problem,Introduction to NP-Hard and NP- Completeness - Decision Trees - P, NP and NP - CompleteProblems - Cook's theorem.

Artificial Intelligence

(Please fill the above with Levels of Correlation, viz., L, M, H)

SYLLABUS:

UNIT – I: Introduction: Introduction to AI, Foundations of AI, History of AI, **Intelligent Agents:** Introduction, Structure of Intelligent Agents, Environments.

UNIT – II: Problem Solving: Problem-Solving Agents, Formulating Problems, Searching for Solutions, Generating action sequences, Data Structures for Search Trees, Search Strategies, **Informed Search Methods:** Best-First Search, Heuristic Functions, Memory Bounded Search, Iterative Improvement Algorithms.

UNIT – III: Agents that Reason Logically: A Knowledge-Based Agent, Wumpus World Environment, Representation, Reasoning, and Logic, Prepositional Logic: Syntax, Semantics, Validity and inference, Agent for the Wumpus World.

R20
urriculum Structure

B. Tech
Computer Science & Engineering

Department of Computer Science & Engineering
University College of Engineering, JNT University Kakinada

First-Order Logic: Syntax and Semantics, Extensions and Notational Variations, Using First-Order Logic, Logical Agents for the Wumpus World, A Simple Reflex Agent.

UNIT – IV: Inference in First-Order Logic: Inference Rules Involving Quantifiers, Generalized Modus Ponens, Forward and Backward Chaining algorithms, Completeness, A Complete Inference Procedure. **Logical Reasoning Systems:** Introduction, Indexing, Retrieval, and Unification, Logic Programming Systems, The Prolog language, Theorem Provers, Design of a theorem prover, Extending Prolog, Forward-Chaining Production Systems,

UNIT – V: Probabilistic Reasoning Systems: Probabilistic Reasoning Systems, The Semantics of Belief Networks, Inference in Belief Networks, Inference in Multiply Connected Belief Networks, Knowledge Engineering for Uncertain Reasoning. **Making Simple Decisions:** Combining Beliefs and Desires Under Uncertainty, Utility Functions, Multi attribute utility functions, Decision Networks, Markov Decision Processes.

Optimization Techniques

Department of Computer Science & Engineering
Malla Reddy College of Engineering, JNTU Hyderabad

SYLLABUS:

UNIT – I: Introduction and Classical Optimization Techniques: Statement of an Optimization problem – design vector – design constraints – constraint surface – objective function – objective function surfaces – classification of Optimization problems.
Classical Optimization Techniques: Single variable Optimization – multi variable Optimization without constraints – necessary and sufficient conditions for minimum/maximum – multivariable Optimization with equality constraints. Solution by method of Lagrange multipliers – multivariable Optimization with inequality constraints – Kuhn – Tucker conditions

UNIT – II: Linear Programming :Standard form of a linear programming problem – geometry of linear programming problems – definitions and theorems – solution of a system of linear simultaneous equations – pivotal reduction of a general system of equations – motivation to the simplex method – simplex algorithm – Duality in Linear Programming – Dual Simplex method.

UNIT – III: Transportation Problem: Finding initial basic feasible solution by north – west corner rule, least cost method and Vogel's approximation method – testing for optimality of balanced transportation problems – Special cases in transportation problem.
UNIT – IV: Nonlinear Programming: Unconstrained cases – One – dimensional minimization methods: Classification, Fibonacci method and Quadratic interpolation method – Univariate method, Powell's method and steepest descent method.
Constrained cases– Characteristics of a constrained problem, Classification, Basic approach of Penalty Function method; Basic approaches of Interior and Exterior penalty function methods, Introduction to convex Programming Problem.

UNIT – V: Dynamic Programming: Dynamic programming multistage decision processes – types – concept of sub optimization and the principle of optimality – computational procedure in dynamic programming – examples illustrating the calculus method of solution – examples illustrating the tabular method of solution.

Data Mining Techniques

UNIT–I: Data Warehousing and Online Analytical Processing: Basic concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, Cloud Data Warehouse, Data Mining and Patten Mining, Technologies, Applications, Major issues, Data Objects & Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity.

UNIT II: Data Preprocessing: An Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization.

UNIT–III: Classification: Basic Concepts, General Approach to solving a classification problem, Decision Tree Induction: Attribute Selection Measures, Tree Pruning, Scalability and Decision Tree Induction, Visual Mining for Decision Tree Induction, Bayesian Classification Methods: Bayes Theorem, Naïve Bayes Classification, Rule-Based Classification, Model Evaluation and Selection.

UNIT–IV: Association Analysis: Problem Definition, Frequent Itemset Generation, Rule Generation: Confident Based Pruning, Rule Generation in Apriori Algorithm, Compact Representation of frequent item sets, FP-Growth Algorithm.

UNIT–V: Cluster Analysis: Overview, Basics and Importance of Cluster Analysis, Clustering techniques, Different Types of Clusters; K-means: The Basic K-means Algorithm, K-means Additional Issues, Bi-secting K Means, Agglomerative Hierarchical Clustering: Basic Agglomerative Hierarchical Clustering Algorithm DBSCAN: Traditional Density Center-Based Approach, DBSCAN Algorithm, Strengths and Weaknesses.

Employability Skills- 1

CO4	Understand the core competencies to succeed in professional environment	K2

based on suggested Revised BTL

UNIT – I:

1. Analytical Thinking & Listening Skills: Self-Introduction, Shaping Young Minds - A Talk by Azim Premji (Listening Activity), Self – Analysis, Developing Positive Attitude, Perception.

2.Communication Skills: Verbal Communication; Non Verbal Communication (Body Language)

UNIT – II:

3.Self-Management Skills: Anger Management, Stress Management, Time Management, Six Thinking Hats, Team Building, Leadership Qualities

4. Etiquette: Social Etiquette, Business Etiquette, Telephone Etiquette, Dining Etiquette

UNIT – III:

5. Standard Operation Methods: Note Making, Note Taking, Minutes Preparation, Email & Letter Writing

6 Verbal Ability: Synonyms, Antonyms, One Word Substitutes-Correction of Sentences-Analogies, Spotting Errors, Sentence Completion, Course of Action -Sentences Assumptions, Sentence Arguments, Reading Comprehension, Practice work

UNIT-IV:

7. Job-Oriented Skills –I: Group Discussion, Mock Group Discussions

8. Job-Oriented Skills –II: Resume Preparation, Interview Skills, Mock Interviews

Compiler Design Lab

List of Experiments:

Experiment 1: Write a C program to identify different types of Tokens in a given Program.
Experiment 2: Write a Lex Program to implement a Lexical Analyzer using Lex tool.
Experiment 3: Write a C program to Simulate Lexical Analyzer to validating a given input String.
Experiment 4: Write a C program to implement the Brute force technique of Top down Parsing.
Experiment 5: Write a C program to implement a Recursive Descent Parser.
Experiment 6: Write C program to compute the *First* and *Follow* Sets for the given Grammar.
Experiment 7: Write a C program for eliminating the left recursion and left factoring of a given grammar

Experiment 8: Write a C program to check the validity of input string using Predictive Parser.

Experiment 9: Write a C program for implementation of LR parsing algorithm to accept a given input string.

R20
Curriculum Structure

B. Tech
Computer Science & Engineering

Department of Computer Science & Engineering
University College of Engineering, JNT University Kakinada

Experiment10: Write a C program for implementation of a Shift Reduce Parser using Stack Data Structure to accept a given input string of a given grammar.

Experiment 11: Simulate the calculator using LEX and YACC tool.
Experiment 12: Generate YACC specification for a few syntactic categories.
Experiment 13: Write a C program for generating the three address code of a given expression/statement.
Experiment 14: Write a C program for implementation of a Code Generation Algorithm of a given expression/statement.

Animation Design

- Designing
- Event and Exhibition Stall Design
- Color Schemes

Software Requirements: Adobe CorelDraw & Illustrator

List of Experiments:

Experiment 1:

1. Design a logo, brochure, cover letter, visiting cards.
2. Prepare a cutout of some images using Photoshop.
3. Place nice background for those images.

Experiment 2:

1. Prepare nice background using gradient tool.
2. Creating Web Banners in Adobe Flash
3. Creating a Logo Animation in Adobe Flash

Experiment 3:

1. Draw Cartoon Animation using reference.
2. Create Lip Sink to Characters
3. Using filters & Special effects

Experiment 4:

1. Create a scene by using Mask layers animation
2. Create any Model of Cars or Bike,
3. Create any model of the male or female character.

Experiment 5:

1. Create a natural outdoor or indoor scene.
2. Render a frame and video of indoor and outdoor scenes.
3. Advance lighting using mental ray render.

Experiment 6:

1. Animate day and night scene of a street with the help of lighting.
2. Title Graphics
3. Video – Audio synchronization
4. 30 Second Commercial AD

Experiment 7:

1. Creating an effect of snow or rain or smoke or water.
2. Creating an effect of bomb/explosion.

Experiment 8:

1. Create a natural outdoor or indoor scene.
2. Set light for Day, Night and Morning
3. Render a frame and video of indoor and outdoor scenes.

Data Mining Techniques with R Lab

C[illegible]	2	2	3	3	3						1	1

[illegible]

Software Requirements: WEKA Tool/Python/R-Tool/Rapid Tool/Oracle Data mining

List of Experiments:

1. Creation of a Data Warehouse.
   - Build Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration Tool, Pentaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects,etc.,)
   - Design multi-dimensional data models namely Star, Snowflake and Fact Constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, manufacturing, Automobiles, sales etc).

   - Write ETL scripts and implement using data warehouse tools.
   - Perform Various OLAP operations such slice, dice, roll up, drill up and pivot

2. Explore machine learning tool "WEKA"
   - Explore WEKA Data Mining/Machine Learning Toolkit.
   - Downloading and/or installation of WEKA data mining toolkit.
   - Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface.
   - Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, Classify panel, Cluster panel, Associate panel and Visualize panel)
   - Study the arff file format Explore the available data sets in WEKA. Load a data set (ex. Weather dataset, Iris dataset, etc.)
   - Load each dataset and observe the following:

     1. List the attribute names and they types

     2. Number of records in each dataset

     3. Identify the class attribute (if any)

     4. Plot Histogram

     5. Determine the number of records for each class.

     6. Visualize the data in various dimensions

3. Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets
   - Explore various options available in Weka for preprocessing data and apply Unsupervised filters like Discretization, Resample filter, etc. on each dataset
   - Load weather. nominal, Iris, Glass datasets into Weka and run Apriori Algorithm with different support and confidence values.
   - Study the rules generated. Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm. Study the rules generated.
   - Derive interesting insights and observe the effect of discretization in the rule generation process.

4. Demonstrate performing classification on data sets Weka/R
   - Load each dataset and run 1d3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic.
   - Extract if-then rules from the decision tree generated by the classifier, Observe the confusion matrix.
   - Load each dataset into Weka/R and perform Naïve-bayes classification and k-Nearest Neighbour classification. Interpret the results obtained.
   - Plot RoC Curves
   - Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset, and deduce which classifier is performing best and poor for each dataset and justify.

5. Demonstrate performing clustering of data sets
   - Load each dataset into Weka/R and run simple k-means clustering algorithm with different values of k (number of desired clusters).
   - Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.
   - Explore other clustering techniques available in Weka/R.
   - Explore visualization features of Weka/R to visualize the clusters. Derive interesting insights and explain.
6. Demonstrate knowledge flow application on data sets into Weka/R
   - Develop a knowledge flow layout for finding strong association rules by using Apriori, FP Growth algorithms
   - Set up the knowledge flow to load an ARFF (batch mode) and perform a cross validation using J48 algorithm
   - Demonstrate plotting multiple ROC curves in the same plot window by using j48 and Random forest tree
7. Demonstrate ZeroR technique on Iris dataset (by using necessary preprocessing technique(s)) and share your observations
8. Write a java program to prepare a simulated data set with unique instances.
9. Write a Python program to generate frequent item sets / association rules using Apriori algorithm
10. Write a program to calculate chi-square value using Python/R. Report your observation.
11. Write a program of Naive Bayesian classification using Python/R programming language.
12. Implement a Java/R program to perform Apriori algorithm
13. Write a R program to cluster your choice of data using simple k-means algorithm using JDK
14. Write a program of cluster analysis using simple k-means algorithm Python/R programming language.
15. Write a program to compute/display dissimilarity matrix (for your own dataset containing at least four instances with two attributes) using Python
16. Visualize the datasets using matplotlib in python/R.(Histogram, Box plot, Bar chart, Pie chart etc.,)