Paper – II: Problem Solving and Python Programming

[4 HPW:: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objectives
The main objective is to teach Computational thinking using Python.

- To know the basics of Programming
- To convert an algorithm into a Python program
- To construct Python programs with control structures.
- To structure a Python Program as a set of functions
- To use Python data structures-lists, tuples, dictionaries.
- To do input/output with files in Python.
- To construct Python programs as a set of objects.

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

1. Develop algorithmic solutions to simple computational problems.
2. Develop and execute simple Python programs.
3. Develop simple Python programs for solving problems.
4. Structure a Python program into functions.
5. Represent compound data using Python lists, tuples, and dictionaries.
6. Read and write data from/to files in Python Programs

Unit-I

Introduction to Computing and Problem Solving: Fundamentals of Computing – Computing Devices – Identification of Computational Problems – Pseudo Code and Flowcharts – Instructions – Algorithms – Building Blocks of Algorithms.
Introduction to Python Programming: Python Interpreter and Interactive Mode– Variables and Identifiers – Arithmetic Operators – Values and Types – Statements, Reading Input, Print Output, Type Conversions, The type() Function and Is Operator, Dynamic and Strongly Typed Language.
Control Flow Statements: The if, The if…else, The if…elif…else Decision Control Statements, Nested if Statement, The while Loop, The for Loop, The continue and break Statements.

Unit-II

Functions: Built-In Functions, Commonly Used Modules, Function Definition and Calling the Function, The return Statement and void Function, Scope and Lifetime of Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command Line Arguments.
Strings: Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings.

Unit-III

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list Parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing - list comprehension; Illustrative programs: selection sort, insertion sort, mergesort, histogram.
Files and exception: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages; Illustrative programs: word count, copy file.

Unit-IV

Object-Oriented Programming: Classes and Objects, Creating Classes in Python, Creating Objects in Python, The Constructor Method, Classes with Multiple Objects, Class Attributes versus Data Attributes, Encapsulation, Inheritance The Polymorphism.
Functional Programming: Lambda. Iterators, Generators, List Comprehensions.

References:

1. Introduction to Python Programming. Gowrishankar S, Veena A. CRC Press, Taylor & Francis Group, 2019
2. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist'', 2nd edition, Updated for Python 3, Shroff/O'Reilly Publishers, 2016 (http://greenteapress.com/wp/think- python/)

Suggested Reading:

1. Learning To Program With Python. Richard L. Halterman. Copyright © 2011
2. Python for Everybody, Exploring Data Using Python 3. Dr. Charles R. Severance. 2016

Practical- 2: Problem Solving and Python Programming (Lab)

[3 HPW: 1 Credit: 25 Marks]

Objective

The main objective of this laboratory is to put into practice computational thinking. The students will be expected to write, compile, run and debug Python programs to demonstrate the usage of

- variables, conditionals and control structures
- functions (both recursive and iterative)
- basic data types as well as compound data structures such as strings, lists, sets, tuples, dictionaries
- object-oriented programming

Exercises

I. Programs to demonstrate the usage of operators and conditional statements

1. Write a program that takes two integers as command line arguments and prints the sum of two integers.
2. Program to display the information:
   Your name, Full Address, Mobile Number, College Name, Course Subjects
3. Program to find the largest number among 'n' given numbers.
4. Program that reads the URL of a website as input and displays contents of a webpage.

II. Programs to demonstrate usage of control structures

5. Program to find the sum of all prime numbers between 1 and 1000.
6. Program that reads set of integers and displays first and second largest numbers.
7. Program to print the sum of first 'n' natural numbers.
8. Program to find the product of two matrices.
9. Program to find the roots of a quadratic equation

III. Programs to demonstrate the usage of Functions and Recursion

10. Write both recursive and non-recursive functions for the following:
    a. To find GCD of two integers
    b. To find the factorial of positive integer
    c. To print Fibonacci Sequence up to given number 'n'
    d. To convert decimal number to Binary equivalent
11. Program with a function that accepts two arguments: a list and a number 'n'. It should display all the numbers in the list that are greater than the given number 'n'.
12. Program with a function to find how many numbers are divisible by 2, 3,4,5,6 and 7 between 1 to 1000

IV. Programs to demonstrate the usage of String functions

13. Program that accept a string as an argument and return the number of vowels and consonants the string contains.
14. Program that accepts two strings S1, S2, and finds whether they are equal are not.
15. Program to count the number of occurrences of characters in a given string.
16. Program to find whether a given string is palindrome or not

V. Programs to demonstrate the usage of lists, sets, dictionaries, tuples and files.

17. Program with a function that takes two lists L1 and L2 containing integer numbers as parameters. The return value is a single list containing the pair wise sums of the numbers in L1 and L2.
18. Program to read the lists of numbers as L1, print the lists in reverse order without Using reverse function.
22. Write a program that combines lists L1 and L2 into a dictionary.
19. Program to find mean, median, mode for the given set of numbers in a list.
20. Program to find all duplicates in the list.
21. Program to o find all the unique elements of a list.
22. Program to find max and min of a given tuple of integers.
23. Program to find union, intersection, difference, symmetric difference of given two sets.
24. Program to display a list of all unique words in a text file
25. Program to read the content of a text file and display it on the screen line wise with a line number followed by a colon
26. Program to analyse the two text files using set operations
27. Write a program to print each line of a file in reverse order.

VI. Programs to demonstrate the usage of Object Oriented Programming

28. Program to implement the inheritance
29. Program to implement the polymorphism

VII. Programs to search and sort the numbers

30. Programs to implement Linear search and Binary search

31. Programs to implement Selection sort, Insertion sort

Paper – III: Data Engineering with Python

[4 HPW:: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objective: The main objective of this course is to teach how to extract raw data, clean the data, perform transformations on data, load data and visualize the data

Outcomes:
At the end of the course the student will be able to:

- Handle different types of files and work with text data
- Use regular expression operations
- Use relational databases via SQL
- Use tabular numeric data
- Use the data structures: data series and frames
- Use PyPlot for visualization

Unit – I

Data Science: Data Analysis Sequence, Data Acquisition Pipeline, Report Structure [Reference 1(Chapter 1-Unit1 to Unit 3)]]
Files and Working with Text Data: Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, The Pickle Module, Reading and Writing CSV Files, Python os and os. Path Modules. [Reference 2, Chapter 9)]
Working with Text Data: JSON and XML in Python[Reference 2, Section12.2]

Unit – II

Working with Text Data: Processing HTML Files, Processing Texts in Natural Languages [Reference 1(Chapter3 –Unit 13, and Unit16)
Regular Expression Operations: Using Special Characters, Regular Expression Methods, Named Groups in Python Regular Expressions, Regular Expression with *glob* Module [Reference 2-Chapter 10]

Unit – III

Working with Databases: Setting Up a MySQL Database, Using a MySQL Database: Command Line, Using a MySQL Database, Taming Document Stores: MongoDB [Reference 1(Chapter4-Unit17toUnit20)]
Working with Tabular Numeric Data(Numpy with Python): NumPy Arrays Creation Using *array()* Function, Array Attributes, NumPy Arrays Creation with Initial Placeholder Content, Integer Indexing, Array Indexing, Boolean Array Indexing, Slicing and Iterating in Arrays, Basic Arithmetic Operations on NumPy Arrays, Mathematical Functions in NumPy, Changing the Shape of an Array, Stacking and Splitting of Arrays, Broadcasting in Arrays. [Reference 2: Section 12.3)]

Unit – IV

Working with Data Series and Frames: Pandas Data Structures, Reshaping Data, Handling Missing Data, Combining Data, Ordering and Describing Data, Transforming Data, Taming Pandas File I/O [Reference 1 (Chapter 6-Unit 31 to Unit 37)]
Plotting: Basic Plotting with PyPlot, Getting to Know Other Plot Types, Mastering Embellishments, Plotting with Pandas [Reference 1(Chapter8-Unit 41 to Unit 44)]

References:

1. Data Science Essentials in Python: Collect, Organize, Explore, Predict, Value. Dmitry Zinoriev,The Pragmatic Programmers LLC, 2016
2. Introduction to Python Programming**.** Gowrishankar S., Veena A. CRC Press, Taylor & Francis Group, 2019

Suggested Reading

3. Python for Everybody: Exploring Data Using Python 3. Charles R Severance, 2016
4. Python Data Analytics – Data Analysis and Science using Pandas, matplotlib and the Python Programming Language. Fabio Nelli, Apress, 2015
5. Website Scraping with Python. Using BeautifulSoup and Scrapy. GáborLászlóHajba, Apress, 2018
6. Machine Learning with Python Cookbook:.Practical Solutions from Preprocessing to Deep Learning. Chris Albon, O'Reilly 2018

Practical- 3: Data Engineering with Python (Lab)

[3 HPW:: 1 Credit :: 25 Marks]

Objective:
The main objective of this laboratory is to put into practice the ETL (extract, transform, load) pipeline which will extract raw data, clean the data, perform transformations on data, load data and visualize the data.

This requires mentoring by TCS.

Libraries
In this course students are expected to extract, transform and load input data that can be text files, CSV files, XML files, JSON, HTML files, SQL databases, NoSQL databases etc.,. For doing this, they should learn the following Python libraries/modules:
pandas, numpy, BeautifulSoup, pymysql, pymongo, nltk, matplotlib

Datasets
For this laboratory, appropriate publicly available datasets, can be studied and used. Example:
MNIST (http://yann.lecun.com/exdb/mnist/),
UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets.html),
Kaggle (https://www.kaggle.com/datasets)
Twitter Data

Exercises

1. Write programs to parse text files, CSV, HTML, XML and JSON documents and extract relevant data. After retrieving data check any anomalies in the data, missing values etc.
2. Write programs for reading and writing binary files
3. Write programs for searching, splitting, and replacing strings based on pattern matching using regular expressions
4. Design a relational database for a small application and populate the database. Using SQL do the CRUD (create, read, update and delete) operations.
5. Create a Python MongoDB client using the Python module pymongo. Using a collection object practice functions for inserting, searching, removing, updating, replacing, and aggregating documents, as well as for creating indexes
6. Write programs to create numpy arrays of different shapes and from different sources, reshape and slice arrays, add array indexes, and apply arithmetic, logic, and aggregation functions to some or all array elements
7. Write programs to use the pandas data structures: Frames and series as storage containers and for a variety of data-wrangling operations, such as:

- Single-level and hierarchical indexing
- Handling missing data
- Arithmetic and Boolean operations on entire columns and tables
- Database-type operations (such as merging and aggregation)
- Plotting individual columns and whole tables
- Reading data from files and writing data to files

Paper – IV: Machine Learning

[4 HPW:: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objectives: The main objective of this course is to teach the principles and foundations of machine learning algorithms

Outcomes:
At the end of the course the student will be able to understand

- Basics of Machine Learning and its limitations
- Machine Learning Algorithms: supervised, unsupervised, bio-inspired
- Probabilistic Modeling and Association Rule Mining

Unit-I

Introduction: What does it mean to learn, Some canonical Learning Problems, The Decision Tree Model of Learning, Formalizing the Learning Problem ID3 Algorithm [Reference1, 2]
Limits of Learning: Data Generating Distributions, Inductive Bias, Not Everything islearnable, Under fitting and Overfitting, Separation of training and test Data, Models, parameters and Hyperparameters, Real World Applications of Machine Learning **Geometry and Nearest Neighbours:** From Data to Feature Vectors, k-Nearest Neighbours, Decision Boundaries, k-means Clustering, High Dimensions [Reference 1]

Unit-II

The Perceptron: Bio-inspired Learning, The Perceptron Algorithm, Geometric Interpretation, Interpreting Perceptron Weights, Perceptron Convergence and Linear Separability, Improved Generalization, Limitations of the Perceptron
Practical Issues: Importance of Good Features, Irrelevant and Redundant Features, Feature Pruning and Normalization, Combinatorial Feature Explosion, Evaluating Model Performance, Cross Validation, Hypothesis Testing and Statistical Significance, Debugging Learning Algorithms, Bias Variance tradeoff
Linear Models: The Optimization Framework for Linear Models, Convex Surrogate Loss Functions, Weight Regularization, Optimization and Gradient Descent, Support Vector Machines [Reference 1]

Unit-III

Probabilistic Modelling: Classification by Density Estimation, Statistical Estimation, Naïve Bayes Models, Prediction [Reference 1]
Neural Networks: Bio-inspired Multi-Layer Networks, The Back-propagation Algorithm, Initialization and Convergence of Neural Networks, Beyond two layers, Breadth vs Depth, Basis Functions [Reference 1]

Unit IV

Unsupervised Learning: Clustering Introduction, Similarity and Distance Measures,Agglomerative Algorithms, Divisive Clustering, Minimum Spanning Tree [Reference 2]
Association Rules: Introduction, large Itemsets, Apriori Algorithm [Reference 2]

References:

1. A Course in Machine Learning (CIML). Hal Daume III, 2017 (freely available online) http://ciml.info/
2. Data Mining: Introductory and Advanced Topics. Margaret H Dunham, Pearson Education, 2003

Suggested Reading:

3. Hands on Machine Learning with SciKit-Learn, Keras and Tensor Flow. AurélienGéron. O'Reily, 2019
4. Machine Learning with Python Cookbook. Chris Albo, O'Reily, 2018
5. Introduction to Machine Learning with Python: A guide. Andreas C Miller, Sarah Guido. O'Reily, 2017

KAKATIYA UNIVERSITY WARANGAL

Under Graduate Courses (Under CBCS AY: 2021-2022 on words)

B.Sc. DATA SCIENCE

II Year: Semester-IV

Practical- 4: Machine Learning (Lab)

[3 HPW:: 1 Credit :: 25 Marks]

Objective:

The main objective of this laboratory is to put into practice the various machine learning algorithms for data analysis using Python and Weka.

ML Toolkits

Students are expected to learn

1. Scikit-learn(https://scikit-learn.org/) an open source machine learning Python library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities.
2. Weka (http://www.cs.waikato.ac.nz/ml/weka/)is another widely used ML toolkit.

Datasets

1. The sklearn datasets package embeds small toy datasets. It includes utilities to load these datasets. It also includes methods to load and fetch popular reference datasets and features some artificial data generators. Students are expected to study and make use of these datasets
2. Weka also has provides various data sets.

References:

1. Scikit-learn user guide.https://scikit-learn.org/stable//_downloads/scikit-learn-docs.pdf
2. Ian Witten, Eibe Frank, and Mark Hall, Chris Pal. DATA MINING: Practical Machine Learning Tools and Techniques, 4th Edition. Morgan Kaufmann.

Exercises

1. Write a Python program using Scikit-learn to split the iris dataset into 70% train data and 30% test data. Out of total 150 records, the training set will contain 120 records and the test set contains 30 of those records. Print both datasets
2. Write Python program to use sklearn's Decision Tree Classifier to build a decision tree for the sklearn's datasets. Implement functions to find the importance of a split (entropy, information gain, gini measure)
3. Write a Python program to implement your own version of the K-means algorithm. Then apply it to different datasets and evaluate the performance.
4. Design a perceptron classifier to classify handwritten numerical digits (0-9). Implement using scikit or Weka.
5. Write a Python program to classify text as spam or not spam using the Naïve Bayes Classifier
6. Use WEKA and experiment with the following classifiers: Association Rule Mining (Apriori), Agglomerative and Divisive Clustering.

Paper – V (A): Natural Language Processing

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objective: The main objective of this course is to give a practical introduction to NLP. It deals with morphological processing, syntactic parsing, information extraction, probabilistic NLP and classification of text using Python's NLTK Library.

Outcomes:

At the end of the course the student will be able to

- Write Python programs to manipulate and analyze language data
- Understand key concepts from NLP and linguistics to describe and analyze language
- Understand the data structures and algorithms that are used in NLP
- Classify texts using machine learning and deep learning

Unit-I

Language Processing and Python: Computing with Language: Texts and Words, A Closer Look at Python: Texts as Lists of Words, Computing with Language: Simple Statistics, Back to Python: Making Decisions and Taking Control, Automatic Natural Language Understanding [Reference 1]

Accessing Text Corpora and Lexical Resources: Accessing Text Corpora, Conditional Frequency Distributions, Lexical Resources, WordNet [Reference 1]

Unit-II

Processing Raw Text: Accessing Text from the Web and from Disk, Strings: Text Processing at the Lowest Level, Text Processing with Unicode, Regular Expressions for Detecting Word Patterns, Useful Applications of Regular Expressions, Normalizing Text, Regular Expressions for Tokenizing Text, Segmentation, Formatting: From Lists to Strings. [Reference 1]

Categorizing and Tagging Words: Using a Tagger, Tagged Corpora, Mapping Words to Properties Using Python Dictionaries, Automatic Tagging, N-Gram Tagging, Transformation-Based Tagging, How to Determine the Category of a Word [Reference 1]

Unit-III

Learning to Classify Text: Supervised Classification, Evaluation, Naive Bayes Classifiers [Reference 1]

Deep Learning for NLP: Introduction to Deep Learning, Convolutional Neural Networks, Recurrent Neural Networks, Classifying Text with Deep Learning [Reference 2]

Unit-IV

Extracting Information from Text

Information Extraction, Chunking, Developing and Evaluating Chunkers, Recursion in Linguistic Structure, Named Entity Recognition, Relation Extraction. [Reference 1]

Analyzing Sentence Structure
Some Grammatical Dilemmas, What's the Use of Syntax. Context-Free Grammar, Parsing with Context-Free Grammar, [Reference 1]

References:

1. Natural Language Processingwith Python. Steven Bird, Ewan Klein, and Edward Lope, O'Reily, 2009
2. Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python. AkshayKulkarni,AdarshaShivananda, Apress, 2019

Suggested Reading:

3. Allen James, Natural Language Understanding, Benjamin/Cumming,1995.
4. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.

Practical – 5(A): Natural Language Processing (Lab)

[3 HPW:: 1 Credit :: 25 Marks]

Objective: The main objective of this laboratory is to write programs that manipulate and analyze language data using Python

This lab requires mentoring sessions from TCS.

Python Packages

Students are expected to know/ learn the following PythonNLP packages

- NLTK (www.nltk.org/ (http://www.nltk.org/))
- Spacy (https://spacy.io/)
- TextBlob (http://textblob.readthedocs.io/en/dev/
- Gensim (https://pypi.python.org/pypi/gensim)
- Pattern (https://pypi.python.org/pypi/Pattern)

Datasets:

1. NLTK includes a small selection of texts from the Project Gutenberg electronic text archive, which contains some 25,000 free electronic books, hosted at *http://www.gutenberg.org/*.
2. The Brown Corpus contains text from 500 sources, and the sources have been categorized by genre, such as *news*, *editorial*, and so on (*http://icame.uib.no/brown/bcm-los.html*).
3. Wikipedia Articles Or any other dataset of your choice

Reference:

Jacob Perkins. Python 3 Text Processing with NLTK 3 Cookbook. Packt Publishing. 2014

Exercises:

1. Text segmentation: Segment a text into linguistically meaningful units, such as paragraphs, sentences, or words. Write programs to segment text (in different formats) into tokens (words and word-like units) using regular expressions. Compare an automatic tokenization with a gold standard
2. Part-of-speech tagging: Label words (tokens) with parts of speech such as noun, adjective, and verb using a variety of tagging methods, e.g., default tagger, regular expression tagger, unigram tagger, and n-gram taggers.
3. Text classification: Categorize text documents into predefined classes using Naïve Bayes Classifier and the Perceptron model
4. Chunk extraction, or partial parsing: Extract short phrases from a part-of-speech tagged sentence. This is different from full parsing in that we're interested in standalone chunks, or phrases, instead of full parse trees

5. Parsing: parsing specific kinds of data, focusing primarily on dates, times, and HTML. Make use of the following preprocessing libraries:
   - dateutil which provides datetime parsing and timezone conversion
   - lxml and BeautifulSoup which can parse, clean, and convert HTML
   - charade and UnicodeDammit which can detect and convert text character encoding
6. Sentiment Analysis: Using Libraries TextBlob and nltk, give the sentiment of a document

(B): NoSQL Data Bases

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objective: The main objective of this course is to cover core concepts of NoSQL databases, along with an example database for each ofthe key-value, document, column family, and graph databases

Outcomes:
At the end of the course the student will be able to

- Understand the need for NoSQL databases and their characteristics
- Understand the concepts of NoSQL databases
- Implement the concepts of NoSQL databases using four example databases: Redis for key-value databases, MongoDB for document databases, Cassandra for column-family databases, and Neo4J for graphdatabases.

Unit-I

Why NoSQL: The Value of Relational Databases, Impedance Mismatch, Application and Integration Databases, Attack of the Clusters, The Emergence of NoSQL
Aggregate Data Models: Aggregates, Column-Family Stores, Summarizing Aggregate-Oriented Databases
More Details on Data Models: Relationships, Graph Databases, Schemaless Databases, Materialized Views, Modeling for Data Access

Unit-II

Distribution Models: Single Server, Sharding, Master-Slave Replication, Peer-to-Peer Replication, Combining Sharding and Replication
Consistency: Update Consistency, Read Consistency, Relaxing Consistency, Relaxing Durability, Quorums
Version Stamps: Business and System Transactions, Version Stamps on Multiple Nodes
Map-Reduce: Basic Map-Reduce, Partitioning and Combining, Composing Map-Reduce Calculations

Unit-III

Key-Value Databases: What Is a Key-Value Store, Key-Value Store Features, Suitable Use Cases, When Not to Use
Document Databases: What Is a Document Database, Features, Suitable Use Cases, When Not to Use

Unit-IV

Column-Family Stores: What Is a Column-Family Data Store, Features, Suitable Use Cases, When Not to Use
Graph Databases: What Is a Graph Database, Features, Suitable Use Cases, When Not to Use

Reference:

1. Pramod J. Sadalage, Martin Fowler. NoSQL Distilled, Addison Wesley 2013

Suggested Reading

2. Luc Perkins, Eric Redmond, Jim R. Wilson. Seven Databases in Seven Weeks. The Pragmatic Bookshelf, 2018
3. Guy Harrison. Next GenerationDatabases: NoSQL, NewSQL, and Big Data. Apress, 2015

Practical – 5(B) : NoSQL Data Bases (Lab)

[3 HPW :: 1 Credit :: 25 Marks]

Objective: The main objective of this lab is to become familiar with the four NoSQL databases: Redis for key-value databases, MongoDB for document databases, Cassandra for column-family databases, and Neo4J for graphdatabases

NoSQL Databases:
Redis (http://redis.io)
MongoDB (http://www.mongodb.org)
Cassandra (http://cassandra.apache.org) Neo4j
(http://neo4j.com)

Exercises:

1. Installation of NoSQL Databases: Redis,MongoDB, Cassandra, Neo4j on Windows & Linux
2. Practice CRUD (*Create*, *Read*, *Update*, and *Delete)* operations on the four databases: Redis,MongoDB, Cassandra, Neo4j
3. Usage of Where Clause equivalent in MongoDB
4. Usage of operations in MongoDB – AND in MongoDB, OR in MongoDB, Limit Records and Sort Records. Usage of operations in MongoDB – Indexing, Advanced Indexing, Aggregation and Map Reduce.
5. Practice with ' macdonalds ' collection data for document oriented database. Import restaurants collection and apply some queries to get specified output.
6. Write a program to count the number of occurrences of a word using MapReduce

Paper – VI - GE: Data Structures and Algorithms

[4 HPW:: 4 Credits :: 100 Marks]

Objectives:

- To introduce the time and space complexities of algorithms.
- To discuss the linear and non-linear data structures and their applications.
- To introduce the creation, insertion and deletion operations on binary search trees and balanced binary searchtrees.
- To introduce various internal sorting techniques and their time complexities

Outcomes:
Students will be

- Able to analyze the time and space complexities of algorithms.
- Able to implement linear, non-linear data structures and balanced binarytrees
- Able to analyze and implement various kinds of searching and sorting techniques.
- Able to find a suitable data structure and algorithm to solve a real world problem.

UNIT–I

Performance and Complexity Analysis: Space Complexity, Time Complexity, Asymptotic Notation (Big-Oh), Complexity Analysis Examples.

Linear List-Array Representation: Vector Representation, Multiple Lists Single Array. **Linear List-Linked Representation:** Singly Linked Lists, Circular Lists, Doubly Linked Lists, Applications (Polynomial Arithmetic).

Arrays and Matrices: Row and Column Major Representations, Sparse Matrices.

Stacks: Array Representation, Linked Representation, Applications (Recursive Calls, Infix to Postfix, Postfix Evaluation).

Queues: Array Representation, Linked Representation. **Skip Lists and Hashing:** Skip Lists Representation, Hash Table Representation, Application- Text Compression.

UNIT- II

Trees: Definitions and Properties, Representation of Binary Trees, Operations, Binary TreeTraversal.
Binary Search Trees: Definitions, Operations on Binary Search Trees.

Balanced Search Trees: AVL Trees, and B-Trees.

UNIT –III

Graphs: Definitions and Properties, Representation, Graph Search Methods (Depth First Search and Breadth First Search)
Application of Graphs: Shortest Path Algorithm (Dijkstra), Minimum Spanning Tree (Prim's and Kruskal's Algorithms).

UNIT –IV

Searching : Linear Search and Binary Search Techniques and their complexity analysis.
Sorting and Complexity Analysis: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, and Heap Sort.Algorithm Design Techniques: Greedy algorithm, divide-and-conquer, dynamic programming.

Suggested Reading:

1. Michael T. Goodrich, Roberto Tamassia, David M. Mount, *Data Structures and Algorithms Python* John Wiley & Sons, 2013.
2. Problem Solving with algorithams and Data Structures Using Python by Miller and David L. Ranum
3. Algorithmic Problem Solving with Python by John B. Schneider

Paper – VII (A): Big Data

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

UNIT – I

Getting an overview of Big Data: Introduction to Big Data, Structuring Big Data, Types of Data, Elements of Big Data, Big Data Analytics, and Advantages of Big Data Analytics.

Introducing Technologies for Handling Big Data: Distributed and Parallel Computing for Big Data, Cloud Computing and Big Data, Features of Cloud Computing, Cloud Deployment Models, Cloud Services for Big Data, Cloud Providers in Big Data Market.

UNIT – II

Understanding Hadoop Ecosystem: Introducing Hadoop, HDFS and MapReduce, Hadoop functions, Hadoop Ecosystem. **Hadoop Distributed File System**- HDFS Architecture, Concept of Blocks in HDFS Architecture, Namenodes and Datanodes, Features of HDFS. MapReduce.

Introducing HBase- HBase Architecture, Regions, Storing Big Data with HBase, Combining HBase and HDFS, Features of HBase, Hive, Pig and Pig Latin, Sqoop, ZooKeeper, Flume, Oozie.

UNIT- III

Understanding MapReduce Fundamentals and HBase: The MapReduceFramework ,Exploring the features of MapReduce, Working of MapReduce, Techniques to optimize MapReduce Jobs, Hardware/Network Topology, Synchronization, File system, Uses of MapReduce, Role of HBase in Big Data Processing- Characteristics of HBase.

Understanding Big Data Technology Foundations: Exploring the Big Data Stack, Data Sources Layer, Ingestion Layer, Storage Layer, Physical Infrastructure Layer, Platform Management Layer, Security Layer, Monitoring Layer, Visualization Layer.

UNIT – IV

Storing Data in Databases and Data Warehouses: RDBMS and Big Data, Issues with Relational Model, Non – Relational Database, Issues with Non Relational Database, Polyglot Persistence, Integrating Big Data with Traditional Data Warehouse, Big Data Analysis and Data Warehouse.

NoSQL Data Management: Introduction to NoSQL, Characteristics of NoSQL, History of NoSQL, Types of NoSQL Data Models- Key Value Data Model, Column Oriented Data Model, Document Data Model, Graph Databases, Schema-Less Databases, Materialized Views, CAP Theorem.

Reference

1. BIG DATA, Black Book TM, DreamTech Press, 2016 Edition.

Suggested Reading:

2. Seema Acharya, SubhasniChellappan , "BIG DATA and ANALYTICS", Wileypublications, 2016
3. Nathan Marz and James Warren, "BIG DATA- Principles and Best Practices of Scalable Real-Time Systems", 2010

Practical – 7(A): Big Data (Lab)

[3 HPW:: 1 Credit :: 25 Marks]

Objectives:

- Installation and understanding of working of HADOOP
- Understanding of MapReduce program paradigm.
- Writing programs in Python using MapReduce
- Understanding working of Pig, Hive
- Understanding of working of Apache Spark Cluster

1. Setting up and Installing Hadoop in its two operating modes:
   - Pseudo distributed,
   - Fully distributed.
2. Implementation of the following file management tasks in Hadoop:
   - Adding files and directories
   - Retrieving files
   - Deleting files
3. Implementation of Word Count Map Reduce program
   - Find the number of occurrence of each word appearing in the input file(s)
   - Performing a MapReduce Job for word search count (look for specific keywords in a file)
4. Map Reduce Program for Stop word elimination:
   - Map Reduce program to eliminate stop words from a large text file.
5. Map Reduce program that mines weather data. Weather sensors collecting data every hour at many locations across the globe gather large volume of log data, which is a good candidate for analysis with MapReduce, since it is semi structured and record-oriented. Data available at: https://github.com/tomwhite/hadoop-book/tree/master/input/ncdc/all.
   - Find average, max and min temperature for each year in NCDC data set?
   - Filter the readings of a set based on value of the measurement, Output the line of input files associated with a temperature value greater than 30.0 and store it in a separate file.
6. Install and Run Pig then write Pig Latin scripts to sort, group, join, project, and filter your data.
7. Write a Pig Latin script for finding TF-IDF value for book dataset (A corpus of eBooks available at: Project Gutenberg)
8. Install and Run Hive then use Hive to create, alter, and drop databases, tables, views, functions, and indexes.
9. Install, Deploy & configure Apache Spark Cluster. Run apache spark applications using Scala.
10. Perform Data analytics using Apache Spark on Amazon food dataset, find all the pairs of items frequently reviewed together.

Paper – VII (B) :Deep Learning

[4 HPW:: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objective: The main objective of this course is to give a practical introduction to DeepLearning using Keras. It covers the concepts of deep learning and their implementation.

Outcomes:

At the end of the course the student will be able to

1. Understand the basics of deep learning
2. Understand the usage of tensors in deep learning
3. Use Python deep-learning framework Keras, with Tensor-Flow as a backend engine.

Unit-I

Introduction: History, Hardware, Data, Algorithms

Neural Networks, Data representations for neural networks, Scalars (0D tensors), Vectors (1D tensors), Matrices (2D tensors), 3D tensors and higher-dimensional tensors, Key attributes,. Manipulating tensors in Numpy, The notion of data batches, Real-world examples of data tensors, Vector data, Time series data or sequence data, Image data, Video data

Unit-II

Tensor operations: Element-wise operations, Broadcasting, Tensor dot, Tensor reshaping, Geometric interpretation of tensor operations, a geometric interpretation of deep learning,

Unit-III

Gradient-based optimization, Derivative of a tensor operation, Stochastic gradient descent,. Chaining derivatives: the Backpropagation algorithm

Neural networks: Anatomy, Layers, Models, Loss functions and optimizers

Unit-IV

Introduction to Keras, Keras, TensorFlow, Theano, and CNTK

Recurrent neural networks: A recurrent layer in Keras, Understanding the LSTM and GRU layers

Reference:

1. FrançoisChollet. Deep Learning with Python. Manning Publications, 2018

Suggested Reading:

2. AurélienGéron. Hands on Machine Learning with SciKit-Learn, Keras and Tensor Flow. O'Reily, 2019
3. Andrew W. Trask. Grokking Deep Learning.Manning Publications, 2019

Practical – 7(B): Deep Learning (Lab)

[3 HPW :: 1 Credit :: 25 Marks]

Objectives: The main objective of this lab is to develop deep learning models using Keras

Deep Learning Tools
Students are expected to learn Keras deep-learning framework (https://keras.io), which is open source and free to download. They should have access to a UNIX machine; though it's possible to use Windows, too. It is also recommended that they work on a recent NVIDIA GPU

Note: The exercises should following **Keras workflow** consisting of four steps

1. Define your training data: input tensors and target tensors
2. Define a network of layers (or *model*) that maps your inputs to your targets
3. Configure the learning process by choosing a loss function, an optimizer, and some metrics to monitor
4. Iterate on your training data by calling the fit() method of your model

Exercise 1:
Dataset:
IMDB dataset, a set of 50,000 highly polarized reviews from the Internet Movie Database. They're split into 25,000 reviews for training and 25,000 reviews for testing, each set consisting of 50% negative and 50% positive reviews. the IMDB dataset comes packaged with Keras
Binary Classification Task:
Build a network to classify movie reviews as positive or negative, based on the text content of the reviews.

Exercise 2:
Dataset:
Reuters dataset, a set of short newswires and their topics, published by Reuters in 1986. It's a simple, widely used toy dataset for text classification. There are 46 different topics; some topics are more represented than others, but each topic has at least 10 examples in the training set. Reuters dataset comes packaged as part of Keras.
Single-label Multi class Classification Task:
Build a network to classify Reuters newswires into 46 mutually exclusive topics. Each data point should be classified into only one category (in this case, topic),. The problem is more specifically an instance of *single-label, multiclass classification*.

Exercise 3:
Dataset:
The Boston Housing Price dataset has an interesting difference from the two previous examples. It has relatively few data points: only 506, split between 404 training samples and 102 test samples. And each *feature* in the input data (for example, the crime rate) has a

different scale. For instance, some values are proportions, which take values between 0 and 1; others take values between 1 and 12, others between 0 and 100, and so on.

Regression Task:

The two previous examples were classification problems, where the goal was to predict a single discrete label of an input data point. Another common type of machine-learning problem is *regression*, which consists of predicting a continuous value instead of a discrete label. You'll attempt to predict the median price of homes in a given Boston suburb in the mid-1970s, given data points about the suburb at the time, such as the crime rate, the local property tax rate, and so on.

4. More exercises can be defined on similar lines.

KAKATIYA UNIVERSITY - WARANGAL - TELANGANA

B.Sc. Programme under CBCS

With effect from the A.Y: 2019

Skill Enhancement Course- IIIII Year

(Common to all Science Courses)

SEMESTER – IV

Fundamentals of Python

Theory: **2 Hours/Week;** **Credits: 2** **Marks: 50 (Internal: 10; External: 40)**

Unit – I

Introduction to Python Programming: How a Program Works, Using Python, Program Development Cycle, Input, Processing, and Output, Displaying Output with the Print Function, Comments, Variables, Reading Input from the Keyboard, Performing Calculations (Operators. Type conversions, Expressions), More about Data Output. Decision Structures and Boolean Logic: if, if-else, if-elif-else Statements, Nested Decision Structures, Comparing Strings, Logical Operators, Boolean Variables. Repetition Structures: Introduction, while loop, for loop, Calculating a Running Total, Input Validation Loops, Nested Loops.Lists and Tuples: Sequences, Introduction to Lists, List slicing, Finding Items in Lists with the in Operator, List Methods and Useful Built-in Functions, Copying Lists, ProcessingLists,

Unit – II

Tuples- operations on tuples, Strings: Basic String Operations, String Slicing, Testing, Searching, and Manipulating Strings. Dictionaries and Sets: Dictionaries, Sets- operations on sets and Dictionaries.
Functions: Introduction, Defining and Calling a Void Function, Designing a Program to Use Functions, Local Variables, Passing Arguments to Functions, Global Variables and Global Constants, Value-Returning Functions- Generating Random Numbers, Writing Our Own Value-Returning Functions, The math Module, Storing Functions in Modules. File and Exceptions: Introduction to File Input and Output, Using Loops to Process Files, Processing Records, Exceptions.

Text Book:

Tony Gaddis, Starting Out With Python (3e)

References:

1. Kenneth A. Lambert, Fundamentals of Python
2. Clinton W. Brownley, Foundations for Analytics with Python
3. James Payne, Beginning Python using Python 2.6 and Python 3
4. Charles Dierach, Introduction to Computer Science using Python
5. Paul Gries, Practical Programming: An Introduction to Computer Science using Python 3

Programming in C **Semester -I**

Theory	4 Hours/Week	4 credit
Practical	3 Hours/Week	1 credit

Unit – I

Computer Fundamentals: Introduction of Computers, Classification of Computers, Anatomy of a Computer, Memory Hierarchy, Introduction to OS, Operational Overview of a CPU.
Program Fundamentals: Generation and Classification of Programming Languages, Compiling, Interpreting, Loading, Linking of a Program, Developing Program, Software Development.
Algorithms: Definitions, Different Ways of Stating Algorithms (Step-form, Pseudo-code, Flowchart), Strategy for Designing Algorithms, Structured Programming Concept.
Basics of C: Overview of C, Developing Programs in C, Parts of Simple C Program, Structure of a C Program, Comments, Program Statements, C Tokens, Keywords, Identifiers, Data Types, Variables, Constants, Operators and Expressions, Expression Evaluation–precedence and associativity, Type Conversions.

Unit – II

Input-Output: Non-formatted and Formatted Input and Output Functions, Escape Sequences,
Control Statements: Selection Statements – if, if-else, nested if, nested if-else, comma operator, conditional operator, switch; Iterative Statements–while, for, do-while; Special Control Statement–goto, break, continue, return, exit.
Arrays and Strings: One-dimensional Arrays, Character Arrays, Functions from ctype.h, string.h, Multidimensional Arrays.

Unit – III

Functions: Concept of Function, Using Functions, Call-by-Value Vs Call-by-reference, Passing Arrays to Functions, Score of Variables, Storage Classes, Inline Functions, and Recursion.
Pointers: Introduction, Address of Operator (&), Pointer, Uses of Pointers, Arrays and Pointers, Pointers and Strings, Pointers to Pointers, Array of Pointers, Pointer to Array, Dynamic Memory Allocation.

Unit – IV

User-defined Data Types: Declaring a Structure (Union) and its members, Initialization Structure (Union), Accessing members of a Structure (Union), Array of Structures (Union), Structures verses Unions, Enumeration Types.
Files: Introduction, Using Files in C, Working with Text Files, Working with Binary Files, Files of Records, Random Access to Files of Records, Other File Management Functions.

Text — Pradip Dey, Manas Ghosh, Computer Fundamentals and Programming in C (2e)

References
BOOKS

Ivor Horton, Beginning C
Ashok Kamthane, Programming in C
Herbert Schildt, The Complete Reference C
Paul Deitel, Harvey Deitel, C How To Program
Byron S. Gottfried, Theory and Problems of Programming with C
Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language
B. A. Forouzan, R. F. Gilberg, A Structured Programming Approach Using C

CHAIRMAN
Board of Studies

C Lab **Semester -I**

Practical 3 Hours/Week 1 credit

1 Write a program to find the largest two (three) numbers using if and conditional operator.

2 Write a program to print the reverse of a given number.

3 Write a program to print the prime number from 2 to n where n is given by user.

4 Write a program to find the roots of a quadratic equation using switch statement.

5 Write a program to print a triangle of stars as follows (take number of lines from user):

```
*
***
*****
*******
*********
```

6 Write a program to find largest and smallest elements in a given list of numbers.

7 Write a program to find the product of two matrices..

8 Write a program to find the GCD of two numbers using iteration and recursion.

9 Write a program to illustrate use of storage classes.

10 Write a program to demonstrate the call by value and the call by reference concepts.

11 Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.

12 Write a program to illustrate use of data type enum.

13 Write a program to demonstrate use of string functions string.h header file.

14 Write a program that opens a file and counts the number of characters in a file.

15 Write a program to create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.

16 Write a program that opens an existing text file and copies it to a new text file with all lowercase letters changed to capital letters and all other characters unchanged.

Note Write the Pseudo Code and draw Flow Chart for the above programs.
Recommended to use Open Source Software: GCC on Linux; DevC++ (or) CodeBlocks on Windows 10.

With Effect from the Academic Year 2019-2020

Programming in C++ Semester -II

Theory	4 Hours/Week	4 credits
Practical	3 Hours/Week	1 credit

Unit – I

Introduction to C++: Applications, Example Programs, Tokens, Data Types, Operators, Expressions, Control Structures, Arrays, Strings, Pointers, Searching and Sorting Arrays.
Functions: Introduction, Prototype, Passing Data by Value, Reference Variables, Using Reference Variables as Parameters, Inline Functions, Default Arguments, Overloading Functions, Passing Arrays to Functions.
Object Oriented Programming: Procedural and Object-Oriented Programming, Terminology, Benefits, OOP Languages, and OOP Applications.

Unit – II

Classes: Introduction, Defining an Instance of a Class, Why Have Private Members? Separating Class Specification from Implementation, Inline Member Functions, Constructors, Passing Arguments to Constructors, Destructors, Overloading Constructors, Private Member Functions, Arrays of Objects, Instance and Static Members, Friends of Classes, Member-wise Assignment, Copy Constructors, Operator Overloading, Object Conversion, Aggregation.

Unit – III

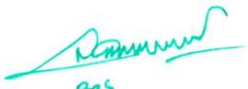
Inheritance: Introduction, Protected Members and Class Access, Base Class Access Specification, Constructors and Destructors in Base and Derived Classes, Redefining Base Class Functions, Class Hierarchies, Polymorphism and Virtual Member Functions, Abstract Base Classes and Pure Virtual Functions, Multiple Inheritance.
C++ Streams: Stream Classes, Unformatted I/O Operations, Formatted I/O Operations.

Unit – IV

Exceptions: Introduction, Throwing an Exception, Handling an Exception, Object-Oriented Exception Handling with Classes, Multiple Exceptions, Extracting Data from the Exception Class, Re-throwing an Exception, Handling the bad_alloc Exception.
Templates: Function Templates–Introduction, Function Templates with Multiple Type, Overloading with Function Templates, Class Templates – Introduction, Defining Objects of the Class Template, Class Templates and Inheritance, Introduction to the STL.

Text — Tony Gaddis, Starting out with C++: from control structures through objects (7e)

References
B. Lippman, C++ Primer
Bruce Eckel, Thinking in C++
K.R. Venugopal, Mastering C++
Herbert Schildt, C++: The Complete Reference
Bjarne Stroustrup, The C++ Programming Language
Sourav Sahay, Object Oriented Programming with C++



CHAIRMAN
Board of Studies
Department of Computer Science
KARAYALAR UNIVERSITY
(T.S.)

With Effect from the Academic Year 2019–2020

C++ Lab Semester -II

Practical 3 Hours/Week 1 credit

1 Write a program to.

a. Print the sum of digits of a given number.

b. Check whether the given number is Armstrong or not

c. Print the prime number from 2 to n where n is natural number given.

2 Write a program to find largest and smallest elements in a given list of numbers and sort the given list.

3 Write a program to read the student name, roll no, marks and display the same using class and object.

4 Write a program to implement the dynamic memory allocation and de-allocation using new and delete operators using class and object.
Write a program to find area of a rectangle, circle, and square using constructors.

5

6 Write a program to implement copy constructor.

7 Write a program using friend functions and friend class.

8 Write a program to implement constructors
§ Default Constructor, Parameterized Constructor, Copy Constructor
§ Define the constructor inside/outside of the class
§ Implement all three constructors within a single class as well as use multiple classes(individual classes)
Write a program to implement the following concepts using class and object
§ Function overloading
§ Operator overloading (unary/binary(+ and -))

Write a program to demonstrate single inheritance, multilevel inheritance and multiple inheritances.

Write a program to implement the overloaded constructors in inheritance.

Write a program to implement the polymorphism and the following concepts using class and object.
§ Virtual functions
§ Pure virtual functions
Write a program to implement the virtual concepts for following concepts
§ Constructor (not applied)
§ Destructor (applied)

Write a program to demonstrate static polymorphism using method overloading.

Write a program to demonstrate dynamic polymorphism using method overriding and dynamic method dispatch.
Write a program to implement the template (generic) concepts
§ Without template class and object
§ With template class and object

Write the Pseudo Code and draw Flow Chart for the above programs.

Recommended to use Open Source Software: GCC on Linux; DevC++ (or) CodeBlocks on Windows.

CHAIRMAN
Board of Studies
Department of Computer Science
KAKATIYA UNIVERSITY
WARANGAL-506 009 (T.S.)

DATA STRUCTURES USING C++

Theory: 4 Hours/Week; **Credits:** 4 **Marks:** 100 (Internal: 20; External: 80)
Practical: 3 Hours/Week **Credits:** 1 **Marks:** 25

Unit - I

Basic data Structure: Introduction to Data Structures, Types of Data Structures, and Introduction to Algorithms, Pseudo code, and Relationship among data, data structures, and algorithms, Implementation of data structures, Analysis of Algorithms.
Stacks: Concept of Stacks and Queues, Stacks, Stack Abstract Data Type, Representation of Stacks Using Sequential Organization (Arrays), Multiple Stacks, Applications of Stack, Expression Evaluation and Conversion, Polish notation and expression conversion, Processing of Function Calls, Reversing a String with a Stack, Recursion.

Unit - II

Recursion: Introduction, Recurrence, Use of Stack in Recursion, Variants of Recursion, Recursive Functions, Iteration versus Recursion.
Queues: Concept of Queues, Queue as Abstract Data Type, Realization of Queues Using Arrays, Circular Queue, Multi-queues, Dequeue, Priority Queue, Applications of Queues,
Linked Lists: Introduction, Linked List, Linked List Abstract Data Type, Linked List Variants, Doubly Linked List, Circular Linked List, Representation of Sparse Matrix Using Linked List, Linked Stack, Linked Queue.

Unit - III

Trees: Introduction, Types of Trees, Binary Tree, Binary Tree Abstract Data Type, Realization of a Binary Tree, Insertion of a Node in Binary Tree, Binary Tree Traversal, Other Tree Operations, Binary Search Tree, Threaded Binary Tree, Applications of Binary Trees.
Searching and Sorting: Search Techniques-Linear Search, Binary Search, Sorting Techniques- Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort, Comparison of All Sorting Methods, Search Trees: Symbol Table, Optimal Binary Search Tree, AVL Tree (Height-balanced Tree).

Unit - IV

Graphs: Introduction, Representation of Graphs, Graph Traversal – Depth First Search, Breadth First Search, Spanning Tree, Prim's Algorithm, Kruskal's Algorithm.
Hashing: Introduction, Key Terms and Issues, Hash Functions, Collision Resolution Strategies, Hash Table Overflow, Extendible Hashing
Heaps: Basic Concepts, Implementation of Heap, Heap as Abstract Data Type, Heap Sort, Heap Applications.

Text books:

1. Varsha H. Patil "Data structures using C++" Oxford University press, 2012
2. M.T. Goodrich, R. Tamassia and D. Mount, Data Structures and Algorithms in C++, John Wiley and Sons, Inc., 2011.

References:

1. Adam Drozdek "Data structures and algorithm in C++" Second edition, 2001
2. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, Introduction to Algorithms,2nd Ed., Prentice-Hall of India, 2006.
3. Robert L. Kruse and A.J. Ryba, Data Structures and Program Design in C++, PrenticeHall, Inc., NJ, 1998.
4. B. Stroupstrup, The C++ Programming Language, Addison Wesley, 2004
5. D.E. Knuth, Fundamental Algorithms (Vol. I), Addison Wesley, 1997

III Semester, DSC 1C

Database Management System

Unit I

Introduction: Database-System Applications, Purpose of Database Systems, View of Data, Database Languages, Relational Databases, Database Design, Architecture, Data Storage and Querying, Transaction Management, Database Database Users and Administrators.

Introduction to the Relational Model: Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, Relational Operations.

Unit II

Database Design and E-R Model: Overview of the Design Process, The Entity-Relationship Model, Constraints, Removing Redundant Attributes in Entity Sets, Entity-Relationship Diagrams, Reduction to Relational Schemas, Entity-Relationship Design Issues, Extended E-R Features, Alternative Notations for Modeling Data, Other Aspects of Database Design.

Relational Database Design: Features of Good Relational Designs, Atomic Domains and First Normal Form, Decomposition Using Functional Dependencies, Functional-Dependency Theory, Decomposition Using Multivalued Dependencies, More Normal Forms, Database-Design Process.

Unit III

Database-System Architectures: Centralized and Client –Server Architectures, Server System Architectures, Parallel Systems, Distributed Systems, Network Types.

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Subqueries, Modification of the Database.

Unit IV

Intermediate SQL: Join Expressions, Views, Transactions, Integrity Constraints, SQL Data Types and Schemas, Authorization.

Advanced SQL: Accessing SQL From a Programming Language, Functions and Procedures, Triggers, Recursive Queries.

Text book:

1. A. Silberschatz, H. Korth and S. Sudarshan, *Database System Concepts*, 6th Ed., Tata McGraw Hill, 2011

References:

1. J. Morrison, M. Morrison and R. Conrad, *Guide to Oracle 10g*, Thomson Learning, 2005.

2. Loney and Koch, *Oracle 10g: The Complete Reference*, Tata McGraw Hill, 2006.

3. David Flanagan, Java Script, *The Definitive Guide*, O'Reilly Media, 2006.

4. Marty Hall, Larry Brown, and Yaakov Chaikin, *Core Servlets and Java Server Pages: Core Technologies* (Vol. II), 2nd Ed., Sun Microsystems Press, 2006.

5. S.K. Singh, *Database Systems Concepts, Design and Applications*, Pearson Education 2006.

6. Spoken Tutorial on "MySQL" as E-resource for Learning:- http://spoken-tutorial.org

Practical: Database Management System

NOTE:

- All the concepts of programs from Text Book including exercises must be practice, execute and write down in the practical record book.
- Faculty must take care about UG standard programs it should be minimum 25 – 30.
- In the external lab examination student has to execute at least three programs with compilation and deployment steps are necessary.
- External Viva-voce is compulsory.

Example programs:

1. Create a database having two tables with the specified fields, to computerize a library system of a Delhi University College.

   LibraryBooks (Accession number, Title, Author, Department, PurchaseDate, Price)
   IssuedBooks (Accession number, Borrower)

   a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
   b) Delete the record of book titled "Database System Concepts".
   c) Change the Department of the book titled "Discrete Maths" to "CS".
   d) List all books that belong to "CS" department.
   e) List all books that belong to "CS" department and are written by author "Navathe".
   f) List all computer (Department="CS") that have been issued.
   g) List all books which have a price less than 500 or purchased between "01/01/1999" and "01/01/2004".

2. Create a database having three tables to store the details of students of Computer Department in your college.

   Personal information about Student (College roll number, Name of student, Date of birth, Address, Marks(rounded off to whole number) in percentage at 10 + 2, Phone number)
   Paper Details (Paper code, Name of the Paper)
   Student's Academic and Attendance details (College roll number, Paper code, Attendance, Marks in home examination).

   a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
   b) Design a query that will return the records (from the second table) along with the name of student from the first table, related to students who have more than 75% attendance and more than 60% marks in paper 2.
   c) List all students who live in "Delhi" and have marks greater than 60 in paper 1.
   d) Find the total attendance and total marks obtained by each student.
   e) List the name of student who has got the highest marks in paper 2.

3. Create the following tables and answer the queries given below:

Customer (CustID, email, Name, Phone, ReferrerID)
Bicycle (BicycleID, DatePurchased, Color, CustID, ModelNo)
BicycleModel (ModelNo, Manufacturer, Style)
Service (StartDate, BicycleID, EndDate)

a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
b) List all the customers who have the bicycles manufactured by manufacturer "Honda".
c) List the bicycles purchased by the customers who have been referred by customer "C1".
d) List the manufacturer of red colored bicycles.
e) List the models of the bicycles given for service.

4. Create the following tables, enter at least 5 records in each table and answer the queries given below.

EMPLOYEE (Person_Name, Street, City)
WORKS (Person_Name, Company_Name, Salary)
COMPANY (Company_Name, City)
MANAGES (Person_Name, Manager_Name)

a) Identify primary and foreign keys.
b) Alter table employee, add a column "email" of type varchar(20).
c) Find the name of all managers who work for both Samba Bank and NCB Bank.
d) Find the names, street address and cities of residence and salary of all employees who work for "Samba Bank" and earn more than $10,000.
e) Find the names of all employees who live in the same city as the company for which they work.
f) Find the highest salary, lowest salary and average salary paid by each company.
g) Find the sum of salary and number of employees in each company.
h) Find the name of the company that pays highest salary.

5. Create the following tables, enter at least 5 records in each table and answer the queries given below.

Suppliers (SNo, Sname, Status, SCity)
Parts (PNo, Pname, Colour, Weight, City)
Project (JNo, Jname, Jcity)
Shipment (Sno, Pno, Jno, Qunatity)

a) Identify primary and foreign keys.
b) Get supplier numbers for suppliers in Paris with status>20.
c) Get suppliers details for suppliers who supply part P2. Display the supplier list in increasing order of supplier numbers.
d) Get suppliers names for suppliers who do not supply part P2.
e) For each shipment get full shipment details, including total shipment weights.
f) Get all the shipments where the quantity is in the range 300 to 750 inclusive.

g) Get part nos. for parts that either weigh more than 16 pounds or are supplied by suppliers S2, or both.
h) Get the names of cities that store more than five red parts.
i) Get full details of parts supplied by a supplier in Delhi.
j) Get part numbers for part supplied by a supplier in Allahabad to a project in Chennai.
k) Get the total number of project supplied by a supplier (say, S1).
l) Get the total quantity of a part (say, P1) supplied by a supplier (say, S1).

Paper- : Programming in Java

Unit I

Introduction**:** Java Essentials, JVM, Java Features, Creation and Execution of Programs, Data Types, Type Conversion, Casting, Conditional Statements, Loops, Branching Mechanism, Classes, Objects, Class Declaration, Creating Objects, Method Declaration and Invocation, Method Overloading,

Unit II

Constructors – Parameterized Constructors, Constructor Overloading, Cleaning-up unused Objects. Class Variables &Method-static Keyword, this Keyword, One-Dimensional Arrays, Two-Dimensional Arrays, Command-Line Arguments, Inner Class.

Inheritance: Introduction, Types of Inheritance, extends Keyword, Examples, Method Overriding, super, final Keyword, Abstract classes, Interfaces, Abstract Classes Verses Interfaces.

Packages: Creating and Using Packages, Access Protection, Wrapper Classes, String Class, String Buffer Class.

Unit III

Exception: Introduction, Types, Exception Handling Techniques, User-Defined Exception.

Multithreading: Introduction, Main Thread and Creation of New Threads –By Inheriting the Thread Class or Implementing the Runnable Interface, Thread Lifecycle, Thread Priority and Synchronization.

Input/Output: Introduction, java.io Package, File Class, FileInputStream Class, FileOutputStream Class, Scanner Class, BufferedInputStream Class, BufferedOutputStream Class, RandomAccessFile Class.

Unit IV

Applets: Introduction, Example, Life Cycle, Applet Class, Common Methods Used in Displaying the Output.

Event Handling: Introduction, Types of Events, Example.

AWT: Introduction, Components, Containers, Button, Label, Checkbox, Radio Buttons, Container Class, Layouts. Swing: Introduction, Differences between Swing and AWT, Jframe, Japplet, Jpanel, Components in Swings, Layout Managers, Jtable, Dialog Box.

Database Handling Using JDBC: Introduction, Types of JDBC Drivers, Load the Driver, Establish Connection, Create Statement, Execute Query, Iterate Resultset, Scrollable Resultset, Developing a JDBC Application.

Sachin Malhotra, Saurabh Choudhary, Programming in Java (2e)

References:

1. Bruce Eckel, Thinking in Java (4e)
2. Herbert Schildt, Java: The Complete Reference (9e)
3. Y. Daniel Liang, Introduction to Java Programming (10e)
4. Paul Deitel, Harvey Deitel, Java: How To Program (10e)
5. Cay S. Horsttnann, Core Java Volume I –Fundamentals (10e)
6. C. Thomas Wu, An introduction to object-oriented programming with Java (5e)
7. Tony Gaddis, Starting Out with Java From Control Structures Through Objects (6e)
8. Jeanne Boyarsky, Scott Selikoff, OCA: Oracle Certified Associate Java SE 8 Programmer–I Study Guide

Programming in Java Lab

Note:

- Programs of all the Concepts from Text Book including exercises must be practice and execute.
- Faculty must take care about UG Standard Programs.
- In the external lab examination student has to execute two programs with compilation and deployment steps are necessary.
- External Vice-Voce is compulsory.

1. Write Java programs to find the following
   a) largest of given three numbers
   b) reverses the digits of a number
   c) given number is prime or not
   d) GCD of given two integers

2. Write Java programs to implement the following
   a) default constructor b) parameterized constructor c) constructor overloading

3. a) Write a Java program to find the smallest from given list of integers using array and scanner class.
   b) Write a Java program for multiplication of two matrices.

4. a) Write a Java program for demonstrating an inner class or nested class.
   b) Write a Java program to implement method overloading, method overriding, dynamic method dispatch

5. Write a Java program to implement single, multilevel, hierarchal, multiple, hybrid inheritances.

6. Write Java programs that demonstrate the use of abstract, this, super, static, final keywords

7. a) Write a Java program for creating a package and using a package.
   b) Write a Java program to demonstrate the use of wrapper classes.

8. a) Write a Java program using all five keywords of exception handling mechanism.

b) Write a Java program for creating customized (user) Exception

9.
a) Write a Java program that checks whether a given string is a palindrome or not.
b) Write a Java program for sorting a given list of names in ascending order.

10.
a) Write a Java program to create a file, write the data and display the data.
b) Write a Java program that reads a file name from user and displays its information.

11.
a) Write a Java program for controlling main thread.
b) Write a Java program for creating new thread by extending Thread class.

12.
a) Write a Java program for creating new thread by implementing Runnable interface.
b) Write a Java program for thread synchronization.

13.
a) Write a Java program to create following AWT components: Button, Checkbox,Choice and List.
b) Write Java programs to create AWT application using containers and layouts.

14.
a) Write Java programs to create a simple Applet.
b) Write a Java program to handle different types of events in a swing application.

15. Write Java programs to create a swing application using swing components and layouts.

16. Write a Java program to store and retrieve data from database using JDBC.

KAKATIYA UNIVERSITY
FACULTY OF SCIENCE
B.Sc. (Computer Science)
SEMESTER – VI
Web Technologies

Theory	4 Hours/Week	4 Credit	Internal marks = 20
Practical	3 Hours/Week	1 Credit	External Marks = 80

Unit – I

Introduction To XHTML– Introduction, first HTML, Headings, Linking, Images, special characters and horizontal rules, Lists, Tables, Frames, Forms, internal linking, meta Elements. CASCADING STYLE SHEETS – Introduction, Inline Styles, Embedded Style Sheets, Conflicting Styles, Linking external sheets, position Elements, box model and text flow, media types, building a CSS drop-down menu, user style sheets, CSS3.

Unit – II

Introduction To Java Scripting- introduction, simple program, prompt dialog and alert boxes, memory concepts, operators, decision making, control structures, if… else statement, while, counter-controlled repetitions, switch statement, do… while statement, *break* and *continue* statements. Functions – program modules in JavaScript, programmer–defined functions, functions definition, scope rules, global functions, Recursion.

Unit – III

Arrays- introduction, declaring and allocating arrays, references and reference parameters, passing arrays to functions. Multidimensional arrays, **EVENTS** – registering event handling, event onload, onmouseover, onmouseout, onfocus, onblur, onsubmit, onreset, event bubbling, more events. **JAVA SCRIPT OBJECTS** – introduction to object technology, Math Object, String Object, Date Object, Boolean and Number Object, document and window Objects, using cookies.

Unit – IV

XML - Introduction, XML Basics, Structuring Data, XML Namespaces, Document Type Definitions (DTDs), W3C XML Schema Documents, XML Vocabularies, Extensible Style sheet Language and XSL Transformations, Document Object Model (DOM).

Ajax-Enabled Rich Internet Applications: introduction, history of Ajax, traditional web applications Vs Ajax Applications, RIAs with Ajax, Ajax example using XMLHttpRequest object, XML and DOM, creating full scale Ajax-enabled application, Dojo Toolkit.

Text Book:

1. Internet & World Wide Web: HOW TO PROGRAM- H. M. Deitel, P.J. Deitel, -Fourth Edition- Pearson edition.

Dr. B. RAMA
Professor
Department of Computer Science
KAKATIYA UNIVERSITY
Warangal- 506 009 (T.S.)