VIKRAMA SIMHAPURI UNIVERSITY::NELLORE

Common Framework of CBCS for Colleges in Andhra Pradesh
(A.P. State of Council of Higher Education)

B.Sc. COMPUTER SCIENCE WITH DATA SCIENCE

Under CBCS Frame Work
(*with effect from the Academic Year 2020-21*)

PROGRAMME: FOUR-YEAR

(With Learning Outcomes, Unit-wise Syllabus, References, Co-curricular Activities & Model Q.P. for Fifteen Courses of 1, 2, 3 & 4 Semesters)

(with effect from 2020-21 Academic Year)

Structure of Computer Science with Data Science

Programme: B.Sc. Computer Science with Data Science as one of the Core Subjects.
Discipline: Computer Science

Year	Semester	Paper Code	Subject	Hours/ Week	Credits	IA	ES	Total
First Year	I	C1	Problem Solving in C	4	4	25	75	100
		C1-P	Problem Solving in C Lab	2	1	--	50	50
	II	C2	Data Structures using C	4	4	25	75	100
		C2-P	Data Structures using C Lab	2	1	--	50	50
Second Year	III	C3	Database Management System	4	4	25	75	100
		C3-P	Database Management System Lab	2	1	--	50	50
	IV	C4	Object Oriented Programming using Java	4	4	25	75	100
		C4-P	Object Oriented Programming using Java Lab	2	1	--	50	50
		C5	Introduction to Data Science with 'R'	4	4	25	75	100
		C5-P	Introduction to Data Science with 'R' Lab	2	1	--	50	50

NOTE :
Syllabus of Paper Codes from C1 to C4 are as same as B.Sc.(Computer Science)

Semester	Course Code	Course Title	Hours	Credits
I	**C1**	**PROBLEM SOLVING IN C**	**60**	**4**

Objectives:

This course aims to provide exposure to problem-solving through programming. It introduces the concepts of the C Programming language.

Course Learning Outcomes:

Upon successful completion of the course, a student will be able to:

1. Understand the evolution and functionality of a Digital Computer.
2. Apply logical skills to analyse a given problem
3. Develop an algorithm for solving a given problem.
4. Understand 'C' language constructs like Iterative statements, Array processing, Pointers, etc.
5. Apply 'C' language constructs to the algorithms to write a 'C' language program.

UNIT- I

General Fundamentals: Introduction to computers: Block diagram of a computer, characteristics and limitations of computers, applications of computers, types of computers, computer generations.

Introduction to Algorithms and Programming Languages: Algorithm – Key features of Algorithms, Flow Charts, Programming Languages – Generations of Programming Languages – Structured Programming Language.

UNIT- II

Introduction to C: Introduction – Structure of C Program – Writing the first C Program – File used in C Program – Compiling and Executing C Programs – Using Comments – Keywords – Identifiers – Basic Data Types in C – Variables – Constants – I/O Statements in C- Operators in C- Programming Examples.

Decision Control and Looping Statements: Introduction to Decision Control Statements– Conditional Branching Statements – Iterative Statements – Nested Loops – Break and Continue Statement – Goto Statement

UNIT- III

Arrays: Introduction – Declaration of Arrays – Accessing elements of the Array – Storing Values in Array– Operations on Arrays – one dimensional, two dimensional, character handling and strings.

UNIT- IV

Functions: Introduction – using functions – Function declaration/ prototype – Function definition – function call – return statement – Passing parameters – Scope of variables – Storage Classes – Recursive functions.

Structure, Union, and Enumerated Data Types: Introduction – Nested Structures – Arrays of Structures – Structures and Functions– Union – Arrays of Unions Variables – Unions inside Structures.

UNIT- V

Pointers: Understanding Computer Memory – Introduction to Pointers – declaring Pointer Variables – Pointer Expressions and Pointer Arithmetic – Null Pointers– Memory Allocation in C Programs – Dynamic Memory Allocation – Drawbacks of Pointers

Files: Introduction to Files – Using Files in C – Reading Data from Files – Writing Data to Files – Detecting the End-of-file.

BOOKS

1. E Balagurusamy – Programming in ANSIC – Tata McGraw-Hill publications.
2. Brain W Kernighan and Dennis M Ritchie - The ‘C’ Programming language” - Pearson publications.
3. Ashok N Kamthane: Programming with ANSI and Turbo C, Pearson Edition Publications.
4. Yashavant Kanetkar - Let Us ‘C’ – BPB Publications.

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and

outside the syllabus content. Shall be individual and challenging)

2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity

B. General

1. Group Discussion
2. Try to solve MCQ's available online.
3. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS:

Some of the following suggested assessment methodologies could be adopted;

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Problem-solving exercises,
4. Practical assignments and laboratory reports,
5. Observation of practical skills,
6. Individual and group project reports like "Creating Text Editor in C".
7. Efficient delivery using seminar presentations,
8. Viva voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs form individual and collaborative work

Semester	Course Code	Course Title	Hours	Credits
I	**C1-P**	**PROBLEM SOLVING IN C LAB**	**30**	**1**

1. Write a program to check whether the given number is Armstrong or not.
2. Write a program to find the sum of individual digits of a positive integer.
3. Write a program to generate the first n terms of the Fibonacci sequence.
4. Write a program to find both the largest and smallest number in a list of integer values
5. Write a program to demonstrate refection of parameters in swapping of two integer values using Call by Value & Call by Address
6. Write a program that uses functions to add two matrices.
7. Write a program to calculate factorial of given integer value using recursive functions
8. Write a program for multiplication of two N X N matrices.
9. Write a program to perform various string operations.
10. Write a program to search an element in a given list of values.
11. Write a program to sort a given list of integers in ascending order.
12. Write a program to calculate the salaries of all employees using ***Employee (ID, Name, Designation, Basic Pay, DA, HRA, Gross Salary, Deduction, Net Salary)*** structure.
    a. DA is 30 % of Basic Pay
    b. HRA is 15% of Basic Pay
    c. Deduction is 10% of (Basic Pay + DA)
    d. Gross Salary = Basic Pay + DA+ HRA
    e. Net Salary = Gross Salary - Deduction
13. Write a program to illustrate pointer arithmetic.
14. Write a program to read the data character by character from a file.
15. Write a program to create ***Book (ISBN,Title, Author, Price, Pages, Publisher)***structure and store book details in a file and perform the following operations
    a. Add book details
    b. Search a book details for a given ISBN and display book details, if available
    c. Update a book details using ISBN
    d. Delete book details for a given ISBN and display list of remaining Books

Semester	Course Code	Course Title	Hours	Credits
II	**C2**	**DATA STRUCTURES USING C**	**60**	**4**

Course Objectives
To introduce the fundamental concept of data structures and to emphasize the importance of various data structures in developing and implementing efficient algorithms.

Course Learning Outcomes: Upon successful completion of the course, a student will be able to:

1. Understand available Data Structures for data storage and processing.
2. Comprehend Data Structure and their real-time applications - Stack, Queue, Linked List, Trees and Graph
3. Choose a suitable Data Structures for an application
4. Develop ability to implement different Sorting and Search methods
5. Have knowledge on Data Structures basic operations like insert, delete, search, update and traversal
6. Design and develop programs using various data structures
7. Implement the applications of algorithms for sorting, pattern matching etc

UNIT – I:
Introduction to Data Structures: Introduction to the Theory of Data Structures, Data Representation, Abstract Data Types, Data Types, Primitive Data Types, Data Structure and Structured Type, Atomic Type, Difference between Abstract Data Types, Data Types, and Data Structures, Refinement Stages

Principles of Programming and Analysis of Algorithms: Program Design, Algorithms, Different Approaches to Designing an Algorithm, Complexity, Big 'O' Notation, Algorithm Analysis, Structured Approach to Programming, Recursion

UNIT – II:
Arrays: Introduction to Linear and Non- Linear Data Structures, One- Dimensional Arrays, Array Operations, Two- Dimensional arrays, Pointers and Arrays, an Overview of Pointers

Linked Lists: Introduction to Lists and Linked Lists, Dynamic Memory Allocation, Basic Linked List Operations, Doubly Linked List, Circular Linked List, Atomic Linked List,

Linked List in Arrays, Linked List versus Arrays

UNIT – III

Stacks: Introduction to Stacks, Stack as an Abstract Data Type, Representation of Stacks through Arrays, Representation of Stacks through Linked Lists, Applications of Stacks, Stacks and Recursion

Queues: Introduction, Queue as an Abstract data Type, Representation of Queues, Circular Queues, Double Ended Queues- Deques, Priority Queues, Application of Queues

UNIT – IV

Binary Trees: Introduction to Non- Linear Data Structures, Introduction Binary Trees, Types of Trees, Basic Definition of Binary Trees, Properties of Binary Trees, Representation of Binary Trees, Operations on a Binary Search Tree, Binary Tree Traversal, Counting Number of Binary Trees, Applications of Binary Tree

UNIT – V:

Searching and sorting: Sorting – An Introduction, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Searching – An Introduction, Linear or Sequential Search, Binary Search.

Graphs: Introduction to Graphs, Terms Associated with Graphs, Sequential Representation of Graphs, Linked Representation of Graphs, Traversal of Graphs, Spanning Trees, Shortest Path, Application of Graphs.

BOOKS:

1. "Data Structures using C", ISRD group Second Edition, TMH
2. "Data Structures through C", YashavantKanetkar, BPB Publications
3. "Data Structures Using C" Balagurusamy E. TMH

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))

4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity))

B. General

1. Group Discussion
2. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS: Some of the following suggested assessment methodologies could be adopted:

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Programming exercises,
4. Practical assignments and laboratory reports,
5. Observation of practical skills,
6. Individual and group project reports.
7. Efficient delivery using seminar presentations,
8. Viva voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs form individual and collaborative work

Semester	Course Code	Course Title	Hours	Credits
II	**C2-P**	**DATA STRUCTURES USING C LAB**	**30**	**1**

1. Write a program to read 'N' numbers of elements into an array and also perform the following operation on an array
    a. Add an element at the beginning of an array
    b. Insert an element at given index of array
    c. Update a element using a values and index
    d. Delete an existing element
2. Write a program using stacks to convert a given either one of the following
    a. postfix expression to prefix
    b. prefix expression to postfix
    c. infix expression to postfix
3. Write Programs to implement the Stack operations using an array
4. Write Programs to implement the Stack operations using Liked List.
5. Write Programs to implement the Queue operations using an array.
6. Write Programs to implement the Queue operations using Liked List.
7. Write a program for arithmetic expression evaluation.
8. Write a program for Binary Search Tree Traversals
9. Write a program to implement dequeue using a doubly linked list.
10. Write a program to search an item in a given list using the following Searching Algorithms
    a. Linear Search
    b. Binary Search.
11. Write a program for implementation of the following Sorting Algorithms
    a. Bubble Sort
    b. Quick Sort
12. Write a program for implementation of the following Sorting Algorithms
    a. Insertion Sort
    b. Merge Sort
13. Write a program for polynomial addition using single linked list
14. Write a program to implement Depth First Search graph traversals algorithm
15. Write a program to implement Breadth First Search graph traversals algorithm

Semester	Course Code	Course Title	Hours	Credits
III	**C3**	**DATABASE MANAGEMENT SYSTEMS**	**60**	**4**

Course Objective:

The objective of the course is to introduce the design and development of databases with special emphasis on relational databases.

Course Learning Outcomes: On completing the subject, students will be able to:

1. Gain knowledge of Database and DBMS.
2. Understand the fundamental concepts of DBMS with special emphasis on relational data model.
3. Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database
4. Model database using ER Diagrams and design database schemas based on the model.
5. Create a small database using SQL.
6. Store, Retrieve data in database.

UNIT- I

Overview of Database Management System: Introduction to data, information, database, database management systems, file-based system, Drawbacks of file-Based System, database approach, Classification of Database Management Systems, advantages of database approach, Various Data Models, Components of Database Management System, three schema architecture of database

UNIT- II

Entity-Relationship Model: Introduction, the building blocks of an entity relationship diagram, classification of entity sets, attribute classification, relationship degree, relationship classification, reducing ER diagram to tables - advantages of ER modeling.

UNIT- III

Relational Model: Introduction, CODD Rules, relational data model, concept of key, relational integrity, relational algebra, relational algebra operations, advantages of relational algebra, limitations of relational algebra, Functional dependencies and normal forms upto 3rd normal form and BCNF.

UNIT- IV

Structured Query Language: Introduction, Commands in SQL, Data Types in SQL, SQL Operators, Data Definition Language, Selection Operation, Projection Operation, Aggregate functions, Data Manipulation Language, Table Modification Commands, Join Operation, Set Operations, View, Sub Query.

UNIT- V

PL/SQL: Introduction, Shortcomings of SQL, Structure of PL/SQL, PL/SQL Language Elements, Data Types, Operators Precedence, Control Structure, Steps to Create a PL/SQL, Program, Iterative Control, Procedure, Function, Database Triggers, Types of Triggers.

BOOKS:

1. Database System Concepts by Abraham Silberschatz, Henry Korth, and S. Sudarshan, McGrawhill
2. Database Management Systems by Raghu Ramakrishnan, McGrawhill
3. Principles of Database Systems by J. D. Ullman
4. Fundamentals of Database Systems by R. Elmasri and S. Navathe
5. SQL: The Ultimate Beginners Guide by Steve Tale.

RECOMMENDED CO-CURRICULAR ACTIVITIES: (Co-curricular activities shall not promote copying from textbook or from others work and shall encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))

3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity)

B. General

1. Group Discussion
2. Try to solve MCQ's available online.
3. Others

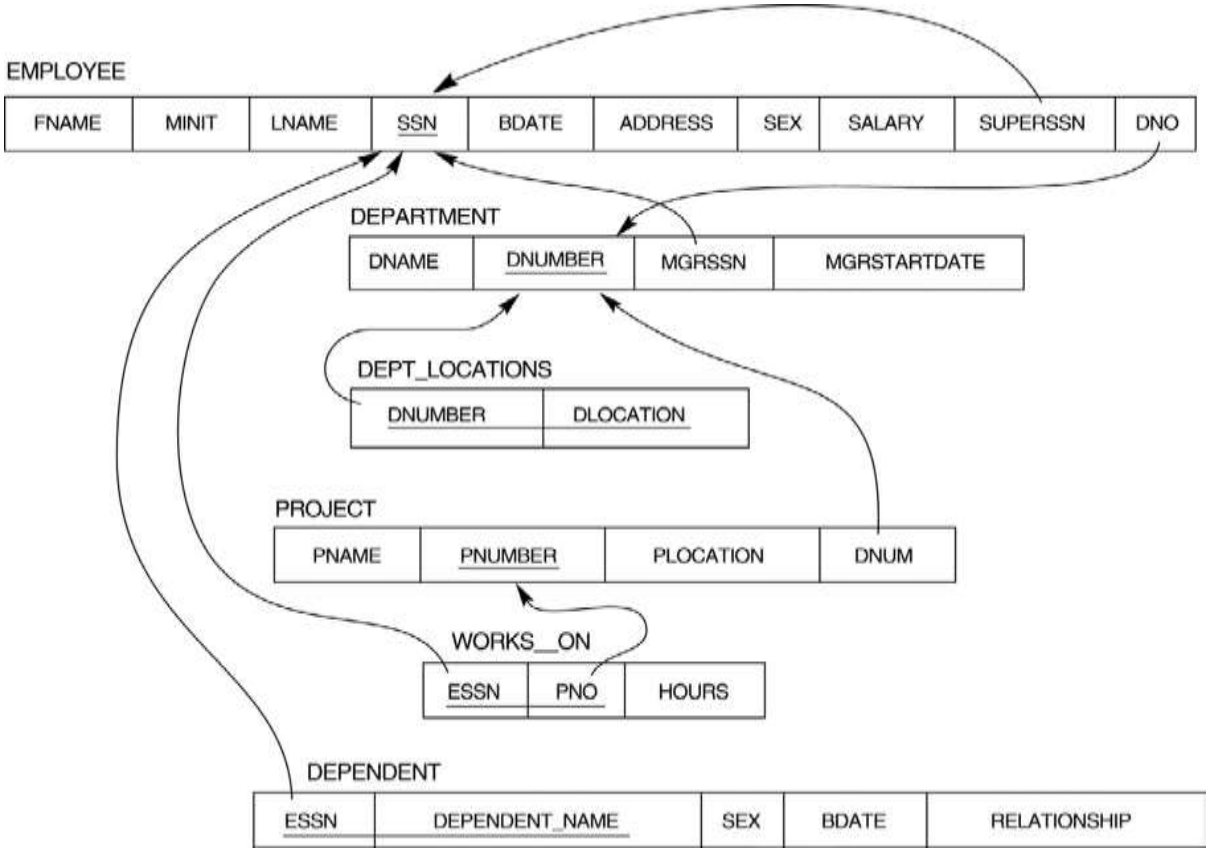
RECOMMENDED CONTINUOUS ASSESSMENT METHODS: Some of the following suggested assessment methodologies could be adopted:

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Practical assignments and laboratory reports,
4. Observation of practical skills,
5. Individual and group project reports like Create your college database for placement purpose.
6. Efficient delivery using seminar presentations,
7. Viva voce interviews.
8. Computerized adaptive testing, literature surveys and evaluations,
9. Peers and self-assessment, outputs form individual and collaborative work

Semester	Course Code	Course Title	Hours	Credits
III	**C3-P**	**DATABASE MANAGEMENT SYSTEMS LAB**	**30**	**1**

1. Draw ER diagram for hospital administration
2. Creation of college database and establish relationships between tables
3. Relational database schema of a company is given in the following figure.

Relational Database Schema - COMPANY



Questions to be performed on above schema

1. Create above tables with relevant ***Primary Key, Foreign Key and other constraints***

2. Display all the details of all employees working in the company.
3. Display ***ssn, lname, fname, address*** of employees who work in department no 7.
4. Retrieve the ***Birthdate and Address*** of the employee whose name is 'Franklin T. Wong'

5. Retrieve all distinct salary values
6. Retrieve all employee names whose address is in 'Bellaire'
7. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000(inclusive)
8. Retrieve the names of all employees who do not have supervisors
9. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date.
10. For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.
11. Retrieve all combinations of Employee Name and Department Name
12. Make a list of all project numbers for projects that involve an employee whose last name is 'Narayan' either as a worker or as a manager of the department that controls the project.
13. Increase the salary of all employees working on the 'ProductX' project by 15%. Retrieve employee name and increased salary of these employees.
14. Retrieve a list of employees and the project name each works in, ordered by the employee's department, and within each department ordered alphabetically by employee first name.
15. Select the names of employees whose salary does not match with salary of any employee in department 10.
16. Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston, or Stafford.
17. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary. Display with proper headings.
18. Find the sum of the salaries and number of employees of all employees of the 'Marketing' department, as well as the maximum salary, the minimum salary, and the average salary in this department.
19. Select the names of employees whose salary is greater than the average salary of all employees in department 10.
20. Delete all dependents of employee whose ***ssn is '123456789'***.
21. Perform a query using alter command to drop/add field and a constraint in Employee table.
22. Create a Function in PL/SQL to find factorial of given number

23. Create a Stored Procedure to increase salary of all employees by 3% and display their modified details

Semester	Course Code	Course Title	Hours	Credits
IV	**C4**	**OBJECT ORIENTATED PROGRAMMING THROUGH JAVA**	**60**	**4**

Objectives:
To introduce the fundamental concepts of Object-Oriented programming and to design & implement object oriented programming concepts in Java.

Course Learning Outcomes: At the end of this course student will:

1. Understand the benefits of a well-structured program
2. Understand different computer programming paradigms
3. Understand underlying principles of Object-Oriented Programming in Java
4. Develop problem-solving and programming skills using OOP concepts
5. Develop the ability to solve real-world problems through software development in high-level programming language like Java

UNIT – I

Introduction to Java: Features of Java, The Java virtual Machine, Parts of Java. **Naming Conventions and Data Types:** Naming Conventions in Java, Data Types in Java, Literals, Type Casting**. Operators in Java:** Operators, Priority of Operators. **Control Statements in Java:** if... else Statement, do... while Statement, while Loop, for Loop, switch Statement, break Statement, continue Statement, return Statement. **Input and Output:** Accepting Input from the Keyboard, Reading Input with Java.util.Scanner Class, Displaying Output with System.out.printf(), Displaying Formatted Output with String.format(). **Arrays:** Types of Arrays, Three Dimensional Arrays (3D array), arrayname.length, Command Line Arguments.

UNIT – II

Strings: Creating Strings, String Class Methods, String Comparison, Immutability of Strings. **Introduction to OOPs:** Problems in Procedure Oriented Approach, Features of Object-Oriented Programming System (OOPS). **Classes and Objects:** Object Creation, Initializing the Instance Variables, Access Specifiers, Constructors. **Methods in Java:** Method Header or Method Prototype, Method Body, Understanding Methods, Static Methods, Static Block, The keyword ‘this’, Instance Methods, Passing Primitive Data Types to Methods, Passing

Objects to Methods, Passing Arrays to Methods, Recursion, Factory Methods

Inheritance: Inheritance, The keyword 'super', The Protected Specifier, Types of Inheritance

UNIT – III

Polymorphism: Polymorphism with Variables, Polymorphism using Methods, Polymorphism with Static Methods, Polymorphism with Private Methods, Polymorphism with Final Methods, final Class. **Abstract Classes:** Abstract Method and Abstract Class. **Interfaces:** Interface, Multiple Inheritance using Interfaces. **Packages:** Package, Different Types of Packages, The JAR Files, Interfaces in a Package, Creating Sub Package in a Package, Access Specifiers in Java, Creating API Document **Exception Handling:** Errors in Java Program, Exceptions, throws Clause, throw Clause, Types of Exceptions, Re – throwing an Exception

UNIT – IV

Threads: Single Tasking, Multi Tasking, Uses of Threads, Creating a Thread and Running it, Terminating the Thread, Single Tasking Using a Thread, Multi Tasking Using Threads, Multiple Threads Acting on Single Object, Thread Class Methods, Deadlock of Threads,

Thread Communication, Thread Priorities, thread Group, Daemon Threads, Applications of Threads, Thread Life Cycle

UNIT – V

Applets: Creating an Applet, Uses of Applets, <APPLET> tag, A Simple Applet, An Applet with Swing Components, Animation in Applets, A Simple Game with an Applet, Applet Parameters

Java Database Connectivity: Database Servers, Database Clients, JDBC (Java Database Connectivity), Working with Oracle Database, Registering the Driver, Connecting to a Database, Preparing SQL Statements, Using jdbc–odbc Bridge Driver to Connect to Oracle Database, Stored Procedures and CallableStatements, Types of Result Sets

BOOKS:

1. Core Java: An Integrated Approach, Authored by Dr. R. Nageswara Rao &Kogent Learning Solutions Inc.
2. E.Balaguruswamy, Programming with JAVA, A primer, 3e, TATA McGraw- Hill Company.

3. John R. Hubbard, Programming with Java, Second Edition, Schaum's outline Series, TMH.
4. Deitel & Deitel. Java TM: How to Program, PHI (2007)

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity

B. General

1. Group Discussion
2. Try to solve MCQ's available online.
3. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS:

Some of the following suggested assessment methodologies could be adopted;

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Programming exercises,
4. Practical assignments and laboratory reports,
5. Observation of practical skills,
6. Individual and group project reports.
7. Efficient delivery using seminar presentations,
8. Viva voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs form individual and collaborative work

Semester	Course Code	Course Title	Hours	Credits
IV	**C4-P**	**OBJECT ORIENTATED PROGRAMMING THROUGH JAVA LAB**	**30**	**1**

1. Write a program to read ***Student Name, Reg.No, Marks[5]*** and calculate ***Total***, ***Percentage, Result***. Display all the details of students
2. Write a program to perform the following String Operations
   a. Read a string
   b. Find out whether there is a given substring or not
   c. Compare existing string by another string and display status
   d. Replace existing string character with another character
   e. Count number of works in a string
3. Java program to implements Addition and Multiplication of two N X N matrices.
4. Java program to demonstrate the use of Constructor.
5. Calculate area of the following shapes using method overloading.
   a. Triangle
   b. Rectangle
   c. Circle
   d. Square
6. Implement inheritance between ***Person (Aadhar, Surname, Name, DOB, and Age)*** and ***Student (Admission Number, College, Course, Year)***classes where ReadData(), DisplayData() are overriding methods.
7. Java program for implementing Interfaces
8. Java program on Multiple Inheritance.
9. Java program for to display ***Serial Number from 1 to N*** by creating two Threads
10. Java program to demonstrate the following exception handlings
    a. Divided by Zero
    b. Array Index Out of Bound
    c. Arithmetic Exception
    d. User Defined Exception

11. Create an Applet to display different shapes such as Circle, Oval, Rectangle, Square and Triangle.

12. Write a program to create ***Book (ISBN,Title, Author, Price, Pages, Publisher)*** table and perform the following operations

a. Add book details

b. Search a book details for a given ISBN and display book details, if available

c. Update a book details using ISBN

d. Delete book details for a given ISBN and display list of remaining Books

Semester	Course Code	Course Title	Hours	Credits
IV	**C5**	**INTRODUCTION TO DATA SCIENCE WITH 'R'**	**60**	**4**

Course Objectives:

The course covers practical issues in statistical computing which includes programming in R, reading data into R, accessing R packages, writing R functions, debugging, and organizing and commenting R code. Topics in statistical data analysis and optimization will provide working examples.

Course Learning Outcomes:
Upon successful completion of the course, a student will be able to:

1. be able to use and program in the programming language R
2. be able to use R to solve statistical problems
3. be able to implement and describe Monte Carlo the technology
4. be able to minimize and maximize functions using R

UNIT – I

Introduction to Data Science: Definition, benefits and uses of data science
Facets of Data: Structured data, unstructured data, natural language, machine generated data, network data, audio, images and video streaming data.
Introduction to R: R overview and history, Basic features of R, Benefits of R, data types in R, Installing R, Getting started with the RStudio IDE, Running R, Packages in R, variable names and assignment, operators, Input/output functions, reading and writing data.

UNIT-II

Preview of Some Important R Data Structures: Vectors, Character Strings, Matrices, Lists, Data Frames, and Classes.
Control structures: Conditional statements, Loops, dates and times, functions, String manipulations.

UNIT-III

Vectors: Scalars, Vectors, Arrays and Matrices: Adding and Deleting Vector Elements, Obtaining the Length of a Vector- **Common vector operations:** Arithmetic & logical operations, Vector Indexing, Generating vector sequences with seq (), Repeating vector constants with rep (), using all () and any () functions, Vectorized operations, NA and NULL values.

UNIT-IV

Matrices and Arrays: Creating Matrices, General Matrix operations- linear algebra operations, matrix indexing, filtering on matrices, using apply() function , Add and Delete matrix rows and columns.
Lists: Creating Lists, General List Operations, List Indexing Adding and Deleting List Elements, Getting the Size of a List **,**Accessing List Components and Values, Using lapply() and sapply() functions.

UNIT-V

Data Frames: Creating Data Frames, Accessing Data Frames - Other Matrix-Like Operations: Extracting sub data frames, using rbind () and cbind () functions.
Factors and Tables : Factors and Levels - Common Functions Used with Factors : tapply() , split() and by() - Working with Tables, Matrix/Array-Like Operations on Tables, Extracting a Sub table - Math Functions: aggregate() and cut() functions.

TEXT BOOKS:

1. The Art of R Programming by Norman Matlof, No starch press, SAN FRANSISCO, 2011.
2. An Introduction to R for Beginners by SASHA HAFNER, on AUG-2019

REFERENCE BOOKS:

1. R Programming for Dummies, Andrie de Vries and Joris Meys, Wiley
2. R for Data Science, Hadley Wickham, Garrett Grolemund, O'Reilly Media
3. R Programming : A Step-By-Step Guide for Absolute Beginners-2nd Edition, Daniel Daniel Bell
4. Learn R programming in 1 Day, Krishna Rungta, Published by Guru99

RECOMMENDED CO-CURRICULAR ACTIVITIES:

(Co-curricular activities shall not promote copying from textbook or from others work and shall encourage self/independent and group learning)

A. Measurable

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity

A. General

1. Group Discussion
2. Try to solve MCQ's available online.
3. Others

RECOMMENDED CONTINUOUS ASSESSMENT METHODS: Some of the following suggested assessment methodologies could be adopted;

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Programming exercises,
4. Practical assignments and laboratory reports,
5. Observation of practical skills,
6. Individual and group project reports.
7. Efficient delivery using seminar presentations,
8. Viva-Voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs form individual and collaborative work

Semester	Course Code	Course Title	Hours	Credits
IV	**C5P**	**INTRODUCTION TO DATA SCIENCE WITH 'R' LAB**	**30**	**1**

1) Write an R-Program to take input from user.
2) Write an R-Program to demonstrate working with operators (Arithmetic, Relational, Logical, Assignment operators).
3) Write an R Program to Check if a Number is Odd or Even
4) Write an R Program to check if the given Number is a Prime Number
5) Write an R Program to Find the Factorial of a Number
6) Write an R Program to Find the Fibonacci sequence Using Recursive Function
7) Write an R Program to create a Vector and to access elements in a Vector
8) Write an R Program to create a Matrix from a Vector using dim() function.
9) Write an R Program to create a List and modify its components.
10) Write an R Program to create a Data Frame.
11) Write an R Program to access a Data Frame like a List.
12) Write an R Program to create a Factor.

VIKRAMA SIMHAPURI UNIVERSITY :: NELLORE
B.SC(COMPUTER SCIENCE) - MODEL QUESTION PAPER

Time: 3 hrs. Max Marks: 75

SECTION – A	**5 X5 = 25 Marks**
*Answer any **FIVE** of the following Questions. Each question carries 5 marks.*	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
SECTION – B	**5 X10 = 50 Marks**
*Answer any **FIVE** of the following Questions. Each question carries 10 marks.*	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	