



VELS

INSTITUTE OF SCIENCE TECHNOLOGY & ADVANCED STUDIES (VISTAS)
(Deemed to be University under section 3 of UGC Act, 1956)
Pallavaram - Chennai



NAAC ACCREDITED WITH 'A' GRADE
Marching Beyond 25 Years Successfully

**Bachelor of
Computer Applications
(B.C.A)**

**Specialization In
Data Science (DS)**

CURRICULUM AND SYLLABUS
(Based on Choice Based Credit System)

Effective from the Academic year

2019 - 2022

DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING SCIENCES

Syllabus

COURSE OUTCOME (Employability)

- To learn the basics of functional hierarchical code organization.
- Ability to define and manage data structures based on problem subject domain.
- Ability to work with characters, pointers and strings.
- Ability to work with arrays of complex objects.
- Understanding a concept of operators and file handling.
- Understanding a defensive programming concept.
- Ability to handle possible errors during program execution.

UNIT I OVERVIEW OF PROGRAMMING 12

Introduction to computer based problem solving, Program design and implementation issues- Flowcharts & Algorithms, Top down design & stepwise refinement, Programming environment – Machine language, assembly language, high level languages, Assemblers, Compilers, Interpreters.

UNIT II FUNDAMENTALS OF C PROGRAMMING 12

Overview of C, Data Types, Constants & Variables, Operators & Expressions, Control constructs-if then, for, while, Arrays- single & multidimensional arrays, Functions-fundamentals – general form, function arguments, return value, Basic I/O-formatted and Unformatted I/O, Advanced features- Type modifiers and storage class specifiers for data types, Bit operators, ? operator, &operator, * operator, Type casting, type conversion.

UNIT III ADVANCED PROGRAMMING TECHNIQUES 12

Control constructs- Do while, Switch statement, break and continue, exit() function, go to and label, Scope rules- Local & global variables, scope rules of functions, Functions-parameter passing, call by value and call by reference, calling functions with arrays, argc and argv, recursion- basic concepts, ex-towers of Hanoi

UNIT IV DYNAMIC DATA STRUCTURES IN C 12

Pointers- The & and * operator, pointer expression, assignments, arithmetic, comparison, malloc vs calloc, arrays of pointers, pointers to pointers, initializing pointers, pointers to functions,

function returning pointers, **Structures**- Basics, declaring, referencing structure elements, array of structures, passing structures to functions, structure pointers, arrays and structures within structures, **Unions** – Declaration, uses, enumerated data-types, typedef.

UNIT V ADDITIONAL FEATURES

12

File Handling – The file pointer, file accessing functions, fopen, fclose, puc, getc, fprintf, C Preprocessor- #define, #include, #undef, Conditional compilation directives, **C standard library and header files**: Header files, string functions, mathematical functions, Date and Time functions

TOTAL HOURS: 60

TEXT BOOK

1. Let us C by Yashwant Kanetka, 6th Edition, PBP Publication

REFERENCE BOOKS

1. The C programming Language by Richie and Kenninghan, 2004, BPB Publication
2. Programming in ANSI C by Balaguruswamy, 3rd Edition, 2005, Tata McGraw Hill.

DATA STRUCTURES AND ALGORITHMS

4 1 0 5

COURSE OUTCOME (**Employability**)

- To **learn the basics of Data Structures**, Searching and sorting techniques, Stack, Queues, Linked List, Trees and graphs and their applications etc.,
- Apply these fundamental principles to solve practical problems related to data structures used for developing applications.

UNIT I INTRODUCTION TO DATA STRUCTURES

12

Definition, **Classification of data structures**: primitive and non-primitive, Elementary data organization, **Time and space complexity of an algorithm** (Examples), String processing. Dynamic memory allocation and pointers: Definition of dynamic memory allocation, Accessing the address of a variable, Declaring and initializing pointers, Accessing a variable through its pointer, Meaning of static and **dynamic memory allocation**, Memory allocation functions: malloc(), calloc(), free() and realloc(). **Recursion**: Definition, Recursion in C (advantages), Writing Recursive programs – Binomial coefficient, Fibonacci, GCD

UNIT II SEARCHING AND SORTING

12

Basic Search Techniques: **Sequential search**: Iterative and Recursive methods, **Binary search**: Iterative and Recursive methods, Comparison between sequential and binary search. Sort: General background and definition, **Bubble sort, Selection sort, Insertion sort, Merge sort, Quick sort.**

UNIT III STACK AND QUEUE

12

Stack – Definition, Array representation of stack, Operations on stack: **Infix, prefix and postfix notations, Conversion of an arithmetic expression from Infix to postfix**, Applications of stacks. **Queue**: Definition, Array representation of queue, Types of queue: Simple queue, Circular queue, Double ended queue (deque), Priority queue, Operations on all types of Queues.

UNIT IV LINKED LIST

12

Definition, Components of linked list, Representation of linked list, Advantages and Disadvantages of linked list. Types of linked list: **Singly linked list, doubly linked list, Circular linked list, Operations on singly linked list: creation, insertion, deletion, search and display.**

UNIT V TREE GRAPHS AND THEIR APPLICATION

12

Definition : Tree, Binary tree, Complete binary tree, **Binary search tree, Heap Tree terminology:**

Root, Node, Degree of a node and tree, Terminal nodes, Non-terminal nodes, Siblings, Level, Edge, Path, depth, Parent node, ancestors of a node. Binary tree: Array representation of tree, Creation of binary tree. Traversal of Binary Tree: Preorder, Inorder and postorder. Graphs, Application of Graphs, Depth First search, Breadth First search.

TOTAL HOURS: 60

TEXT BOOK

1. Weiss, Data Structures and Algorithm Analysis in C, II Edition, Pearson Education, 2001.
2. Lipschutz: Schaum's outline series Data structures Tata McGraw-Hill.
3. Robert Kruse Data Structures and program designing using 'C'.

REFERENCE BOOKS

1. Trembley and Sorenson Data Structures
2. E. Balaguruswamy Programming in ANSI C.
3. Bandyopadhyay, Data Structures Using C Pearson Education, 1999
4. Tenenbaum, Data Structures Using C. Pearson Education, 2006
5. Kamthane: Introduction to Data Structures in C. Pearson Education 2005.
6. Hanumanthappa M., Practical approach to Data Structures, Laxmi Publications, Fire Wall media 2006
7. Langsam, Ausenstein Maoshe & M. Tanenbaum Aaron Data Structures using C and C++ Pearson Education

COURSE OUTCOME (Skill Development)

- Understanding a defensive programming concept.
- Understanding the syntax and semantics of c language- strings, local and global variables.
- Understanding the concepts of pointers and operators.
- Ability to handle possible errors during program execution.

LIST OF EXPERIMENTS**Part A**

- 1 Printing the reverse of an integer.
- 2 Printing the odd and even series of N numbers.
- 3 Get a string and convert the lowercase to uppercase and vice--versa using getchar() and putchar().
- 4 Input a string and find the number of each of the vowels appear in the string.
- 5 Accept N words and make it as a sentence by inserting blank spaces and a full stop at the end.
- 6 Printing the reverse of a string.

Part B

- 1 Searching an element in an array using pointers.
- 2 Checking whether the given matrix is an identity matrix or not.
- 3 Finding the first N terms of Fibonacci series.
- 4 Declare 3 pointer variables to store a character, a character string and an integer respectively. Input values into these variables. Display the address and the contents of each variable.
- 5 Define a structure with three members and display the same.
- 6 Declare a union with three members of type integer, char, string and illustrate the use of union.
- 7 Recursive program to find the factorial of an integer.
- 8 Finding the maximum of 4 numbers by defining a macro for the maximum of two numbers.
- 9 Arranging N numbers in ascending and in descending order using bubble sort.
- 10 Addition and subtraction of two matrices.

- 11 Multiplication of two matrices.
- 12 Converting a hexadecimal number into its binary equivalent.
- 13 Check whether the given string is a palindrome or not.
- 14 Demonstration of bitwise operations.
- 15 Applying binary search to a set of N numbers by using a function.
- 16 Create a sequential file with three fields: empno, empname, empbasic. Print all the details in a neat format by adding 500 to their basic salary.

Total No. of Hours: 60

Course Objective: (Skill Development).

- Office tools course would enable the students in crafting professional word documents, excel spread sheets, power point presentations using the Microsoft suite of office tools.
- To familiarize the students in preparation of documents and presentations with office automation tools

LIST OF EXPERIMENTS

1. Text Manipulation using MS-WORD.
2. Usage of Bullets and Numbering, Header and Footer using MS-WORD.
3. Usage of Spell check, Find & Replace using MS-WORD.
4. Table Manipulation using MS-WORD.
5. Picture Insertion and Alignment using MS-WORD.
6. Usage of Spell check, Find & Replace using MS-WORD.
7. Creation of documents using templates using MS-WORD.
8. Cell Editing using MS-EXCEL.
9. Data Sorting using MS-EXCEL.
10. Usage of Formulas & Built In Functions using MS-EXCEL.
11. Worksheet Preparation using MS-EXCEL.
12. Drawing Graphs using MS-EXCEL.
13. Inserting ClipArt's & Pictures using MS-EXCEL.
14. Slide Transitions and Animation using MS-POWER POINT.
15. Organization Chart using MS-POWER POINT.

COURSE OUTCOME

- This course introduces the basic concepts of UNIX programming instudents knowledge.
- This course acquires the student to write Vi Editor and its usages .

LIST OF EXPERIMENTS

1. Execute 25 basic **commands** of **UNIX**.
2. Basics of **functionality** and **modes** of **VI Editor**.
3. WAP that accepts user name and reports if user is logged in.
4. WAP which displays the following **menu** and **executes the option** selected by user:
1. ls 2. Pwd 3. ls -l 4. ps -fe
5. WAP to print 10 9 8 7 6 5 4 3 2 1 .
6. WAP that replaces all “*.txt” file names with “*.txt.old” in the current.
7. WAP that echoes itself to stdout, but backwards.
8. **WAP that takes a filename as input and checks if it is executable, if not make it executable.**
9. WAP to take string as command line argument and reverse it.
10. **1. Create a data file called employee in the format given below:**
 - a. EmpCode Character
 - b. EmpName Character
 - c. Grade Character
 - d. Years of experience Numeric
 - e. Basic Pay Numeric

```
$vi employee  
A001 ARJUN E1 01 12000.00  
A006 Anand E1 01 12450.00  
A010 Rajesh E2 03 14500.00  
A002 Mohan E2 02 13000.00  
A005 John E2 01 14500.00  
A009 Denial SmithE2 04 17500.00
```

A004 Williams E1 01 12000.00

Perform the following functions on the file:

a. Sort the file on EmpCode.

b. Sort the file on

(i) Decreasing order of basic pay

(ii) Increasing order of years of experience.

c. Display the number of employees whose details are included in the file.

d. Display all records with 'smith' a part of employee name.

e. Display all records with EmpName starting with 'B'.

f. Display the records on Employees whose grade is E2 and have work experience of 2 to 5 years.

g. Store in 'file 1' the names of all employees whose basic pay is between 10000 and 15000.

h. Display records of all employees who are not in grade E2

TOTAL HOURS: 60

Course Outcome: (Skill Development).

- To develop the skills in the areas of Arrays, Sorting techniques, Stack, Queues, Linked List, and Binary Search Tree and Tree Traversing.
- Apply these fundamental principles to solve practical problems related to develop software applications.

LIST OF EXPERIMENTS**Part A**

1. Use a recursive function to find GCD of two numbers.
2. Use a recursive function to find the Fibonacci series.
3. Use pointers to find the length of a string and to concatenate two strings.
4. Use pointers to copy a string and to extract a substring from a given a string.
5. Use a recursive function for the towers of Hanoi with three discs.
6. Insert an integer into a given position in an array.
7. Deleting an integer from an array.
8. Write a program to create a linked list and to display it.
9. Write a program to sort N numbers using insertion sort.
10. Write a program to sort N numbers using selection sort.

Part B

1. Inserting a node into a singly linked list.
2. Deleting a node from a singly linked list.
3. Pointer implementation of stacks.
4. Pointer implementation of queues.
5. Creating a binary search tree and traversing it using in order, preorder and post order.
6. Sort N numbers using merge sort.

Total No of Hours 60