

PAPER CODE – 100104 || 100204

ESC	PROGRAMMING FOR PROBLEM SOLVING	L:3	T:0	P:4	CREDIT:5
------------	--	------------	------------	------------	-----------------

MODULE 1: INTRODUCTION TO PROGRAMMING (6 LECTURES)

INTRODUCTION TO COMPONENTS OF A COMPUTER SYSTEM (DISKS, MEMORY, PROCESSOR, WHERE A PROGRAM IS STORED AND EXECUTED, OPERATING SYSTEM, COMPILERS ETC). IDEA OF ALGORITHM: STEPS TO SOLVE LOGICAL AND NUMERICAL PROBLEMS. REPRESENTATION OF ALGORITHM: FLOWCHART/PSEUDO CODE WITH EXAMPLES. FROM ALGORITHMS TO PROGRAMS; SOURCE CODE, VARIABLES (WITH DATA TYPES) VARIABLES AND MEMORY LOCATIONS, TYPE CASTING/TYPE CONVERSION, RUN TIME ENVIRONMENT (STATIC, DYNAMIC LOCATION), STORAGE CLASSES (AUTO, REGISTER, STATIC, EXTERN), SYNTAX AND LOGICAL ERRORS IN COMPILATION, OBJECT AND EXECUTABLE CODE.

MODULE 2: OPERATORS (3 LECTURES)

ARITHMETIC EXPRESSIONS/ARITHMETIC OPERATORS/RELATIONAL OPERATORS/LOGICAL OPERATORS/BITWISE OPERATORS AND PRECEDENCE

MODULE 3: CONDITIONAL BRANCHING AND LOOPS (5 LECTURES)

WRITING AND EVALUATION OF CONDITIONALS AND CONSEQUENT BRANCHING, ITERATION AND LOOPS

MODULE 4: ARRAYS (4 LECTURES)

ARRAY DECLARATION & INITIALIZATION, BOUND CHECKING ARRAYS (1-D, 2-D), CHARACTER ARRAYS AND STRINGS.

MODULE 5: BASIC ALGORITHMS (6 LECTURES)

SEARCHING (LINEAR SEARCH, BINARY SEARCH ETC.), BASIC SORTING ALGORITHMS (BUBBLE, INSERTION AND SELECTION), FINDING ROOTS OF EQUATIONS, NOTION OF ORDER OF COMPLEXITY THROUGH EXAMPLE PROGRAMS (NO FORMAL DEFINITION REQUIRED)

MODULE 6: FUNCTION (4 LECTURES)

INTRODUCTION & WRITING FUNCTIONS, SCOPE OF VARIABLES FUNCTIONS (INCLUDING USING BUILT IN LIBRARIES), PARAMETER PASSING IN FUNCTIONS, CALL BY VALUE, PASSING ARRAYS TO FUNCTIONS: IDEA OF CALL BY REFERENCE

MODULE 7: RECURSION (5 LECTURES)

RECURSION, AS A DIFFERENT WAY OF SOLVING PROBLEMS. EXAMPLE PROGRAMS, SUCH AS FINDING FACTORIAL, FIBONACCI SERIES, REVERSE A STRING USING RECURSION, AND GCD OF TWO NUMBERS, ACKERMAN FUNCTION ETC. QUICK SORT OR MERGE SORT.

MODULE 8: STRUCTURE/UNION (3 LECTURES)

STRUCTURES, ACCESSING STRUCTURE ELEMENTS, WAY OF STORAGE OF STRUCTURE ELEMENT, DEFINING STRUCTURES AND ARRAY OF STRUCTURES, BASIC DEFINITION OF UNION, COMPARISON B/W STRUCTURE & UNION WITH EXAMPLE

MODULE 9: POINTERS (5 LECTURES)

IDEA OF POINTERS, DEFINING POINTERS, USE OF POINTERS IN SELF-REFERENTIAL STRUCTURES, NOTION OF LINKED LIST (NO IMPLEMENTATION), POINTER TO POINTER, POINTER TO ARRAY, POINTER TO STRINGS, ARRAY OF POINTER, POINTER TO FUNCTION, POINTER TO STRUCTURE.

MODULE 10: FILE HANDLING

(ONLY IF TIME IS AVAILABLE, OTHERWISE SHOULD BE DONE AS PART OF THE LAB)

SUGGESTED TEXT BOOKS

- 📖 BYRON GOTTFRIED, *SCHAUM'S OUTLINE OF PROGRAMMING WITH C*, MCGRAW-HILL
- 📖 E. BALAGURUSWAMY, *PROGRAMMING IN ANSI C*, TATA MCGRAW-HILL

SUGGESTED REFERENCE BOOKS

- 📖 BRIAN W. KERNIGHAN AND DENNIS M. RITCHIE, *THE C PROGRAMMING LANGUAGE*, PRENTICE HALL OF INDIA
- 📖 YASHWANT KANETKAR, *LET US C*, BPB PUBLICATION

THE STUDENT WILL LEARN

- TO FORMULATE SIMPLE ALGORITHMS FOR ARITHMETIC AND LOGICAL PROBLEMS.
- TO TRANSLATE THE ALGORITHMS TO PROGRAMS (IN C LANGUAGE).
- TO TEST AND EXECUTE THE PROGRAMS AND CORRECT SYNTAX AND LOGICAL ERRORS.
- TO IMPLEMENT CONDITIONAL BRANCHING, ITERATION AND RECURSION.
- TO DECOMPOSE A PROBLEM INTO FUNCTIONS AND SYNTHESIZE A COMPLETE PROGRAM USING DIVIDE AND CONQUER APPROACH.
- TO USE ARRAYS, POINTERS AND STRUCTURES TO FORMULATE ALGORITHMS AND PROGRAMS.
- TO APPLY PROGRAMMING TO SOLVE MATRIX ADDITION AND MULTIPLICATION PROBLEMS AND SEARCHING AND SORTING PROBLEMS.
- TO APPLY PROGRAMMING TO SOLVE SIMPLE NUMERICAL METHOD PROBLEMS, NAMELY ROOT FINDING OF FUNCTION, DIFFERENTIATION OF FUNCTION AND SIMPLE INTEGRATION.

LABORATORY PROGRAMMING FOR PROBLEM SOLVING

[THE LABORATORY SHOULD BE PRECEDED OR FOLLOWED BY A TUTORIAL TO EXPLAIN THE APPROACH OR ALGORITHM TO BE IMPLEMENTED FOR THE PROBLEM GIVEN.]

TUTORIAL 1: PROBLEM SOLVING USING COMPUTERS:

LAB1: FAMILIARIZATION WITH PROGRAMMING ENVIRONMENT

TUTORIAL 2: VARIABLE TYPES AND TYPE CONVERSIONS:

LAB 2: SIMPLE COMPUTATIONAL PROBLEMS USING ARITHMETIC EXPRESSIONS

TUTORIAL 3: BRANCHING AND LOGICAL EXPRESSIONS:

LAB 3: PROBLEMS INVOLVING IF-THEN-ELSE STRUCTURES

TUTORIAL 4: LOOPS, WHILE AND FOR LOOPS:

LAB 4: ITERATIVE PROBLEMS E.G., SUM OF SERIES

TUTORIAL 5: 1D ARRAYS: SEARCHING, SORTING:
LAB 5: 1D ARRAY MANIPULATION

TUTORIAL 6: 2D ARRAYS AND STRINGS
LAB 6: MATRIX PROBLEMS, STRING OPERATIONS

TUTORIAL 7: FUNCTIONS, CALL BY VALUE:
LAB 7: SIMPLE FUNCTIONS

TUTORIAL 8: NUMERICAL METHODS (ROOT FINDING, NUMERICAL DIFFERENTIATION,
NUMERICAL INTEGRATION):
LAB 8: PROGRAMMING FOR SOLVING NUMERICAL METHODS PROBLEMS

TUTORIAL 9: RECURSION, STRUCTURE OF RECURSIVE CALLS
LAB 9: RECURSIVE FUNCTIONS

TUTORIAL 10: POINTERS, STRUCTURES AND DYNAMIC MEMORY ALLOCATION
LAB 10: POINTERS AND STRUCTURES

TUTORIAL 11: FILE HANDLING:
LAB 11: FILE OPERATIONS

LABORATORY OUTCOMES

- ❖ TO FORMULATE THE ALGORITHMS FOR SIMPLE PROBLEMS
- ❖ TO TRANSLATE GIVEN ALGORITHMS TO A WORKING AND CORRECT PROGRAM
- ❖ TO BE ABLE TO CORRECT SYNTAX ERRORS AS REPORTED BY THE COMPILERS
- ❖ TO BE ABLE TO IDENTIFY AND CORRECT LOGICAL ERRORS ENCOUNTERED AT RUN TIME
- ❖ TO BE ABLE TO WRITE ITERATIVE AS WELL AS RECURSIVE PROGRAMS
- ❖ TO BE ABLE TO REPRESENT DATA IN ARRAYS, STRINGS AND STRUCTURES AND MANIPULATE THEM THROUGH A PROGRAM
- ❖ TO BE ABLE TO DECLARE POINTERS OF DIFFERENT TYPES AND USE THEM IN DEFINING SELF- REFERENTIAL STRUCTURES.
- ❖ TO BE ABLE TO CREATE, READ AND WRITE TO AND FROM SIMPLE TEXT FILES.
