

First Year B.C.A. (Under Science) Semester II

Course Code: BCA 202

Course Title: Advanced Programming in C

Total Contact Hours: 48 hrs.

Total Credits: 04

Total Marks: 100(60 Lectures)

Teaching Scheme: Theory- 05 Lect./ Week

Course Objectives: The objective of this course is to study the Advanced Programming in C.

UNIT NO.	DESCRIPTION	No. of LECTURES
UNIT 1	1. Preprocessor 1.1. Concept 1.2. Format of preprocessor directives 1.3. File inclusion directives (#include) 1.4. Macro substitution directives (#define), nested macros, parameterized macros 1.5. Macros versus functions 1.6. #error / #pragma directives 1.7. Conditional compilation (#if/#ifdef/#else/#elif/#endif) 1.8. Predefined macros (_DATE_ / _TIME_ / _FILE_ / _LINE_ / _STDC_) 1.9. Preprocessor operators 1.9.1. Macro continuation (\) 1.9.2. stringize (#) 1.9.3. token pasting (##) 1.9.4. defined()	06
UNIT 2	2. Pointers 2.1. Concept – reference & dereference (Data model – Value model v/s Reference model) 2.2. Declaration, definition, initialization & use 2.3. Types of pointers 2.4. Pointer Arithmetic 2.5. Relationship between Arrays & Pointers 2.5.1. Pointer to array 2.5.2. Array of pointers 2.6. Multiple indirection (introduction of double pointer) 2.7. Functions & pointers 2.7.1. Passing pointer to function 2.7.2. Returning pointer from function 2.7.3. Function pointer 2.8. Pointers & const 2.9. Dynamic memory management 2.9.1. Allocation 2.9.2. Resizing 2.9.3. Releasing 2.9.4. Memory leak / dangling pointers	12

UNIT 3	3. Strings 3.1. Concept 3.2. Declaration, definition, initialization, format specifiers 3.3. String literals/ constants & variables – reading & writing from & to console 3.4. Importance of terminating NULL character 3.5. Strings & pointers 3.6. Array of strings & array of character pointers 3.7. User defined functions for predefined functions in string.h 3.7.1. strlen / strcpy / strcat / strcmp / strcmpi / strrev / strlwr /strupr / strset / strchr / strrchr / strstr / strncpy / strncat / strncmp / strncmp / strnset / strtok 3.8. Command line arguments	12
UNIT 4	4. Structures 4.1. Concept 4.2. Declaration, definition, initialization, accessing structure members (. operator) 4.3. Array of structures 4.4. Pointers to structures 4.4.1. Declaring pointer to structure 4.4.2. Accessing structure members via pointer to structure (* operator) 4.5. Structures & functions 4.5.1. Passing each member of structure as a separate argument 4.5.2. Passing structure by value / address 4.6. Nested structures 4.7. typedef& structures 4.8. typedef versus #define 4.9. Bit fields 4.9.1. Concept, need, use	14
UNIT 5	5. Unions 5.1. Concept 5.2. Declaration, definition, accessing union members 5.3. Difference between Structures & unions	06
UNIT 6	6. File Handling 6.1. Concept of streams, filesneed 6.2. Types of files 6.3. Operations on text & binary files 6.4. Random access to files	10

Reference Books:

1. The C Programming Language (Second Edition) – By B. W. Kerninghan& D. M. Ritchie
2. Programming in C – A Practical Approach – By Ajay Mittal (Pearson Publications)
3. Programming with C – By Byron S Gottfried (Schaum's Outlines)
4. A structural Programming Approach using C – By BehrouzForouzan& Richard Gilberg