



Subject: Advanced C Programming -BCAXX12304

Type of course: Major Core

Prerequisite: Basic knowledge of C programming

Rationale:

In the previous semester students have learned procedure for developing programs to solve simple problems using basic features of C language. This course deals with some advanced features of the 'C' language. The objective of this course is to enable you to use advanced C programming language constructs and techniques to create more structured and portable code.

This course is designed to take your basic C skills to the next level and help you obtain mastery of the language by helping you understand advanced concepts of the C programming language, enabling you to master the art of problem-solving in programming using efficient, proven methods.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks					Total Marks
CI	T	P		C	SEE		CCE		
			Theory		Practical	MSE	LWA	ALA	
3	0	2	4	75	25	30	20	50	200

Legends: CI-Class Room Instructions; T – Tutorial; P - Practical; C – Credit; SEE - Semester End Evaluation; MSE- Mid Semester Examination; LWA - Lab Work Assessment; V – Viva voce; CCE- Continuous and Comprehensive Evaluation; ALA- Active Learning Activities.

3 Credits * 25 Marks = 75 Marks (each credit carries 25 Marks) Theory

1 Credits * 25 Marks = 25 Marks (each credit carries 25 Marks) Practical

SEE 100 Marks will be converted in to 50 Marks

CCE 100 Marks will be converted in to 50 Marks

It is compulsory to pass in each individual component.



CourseContent:

Sr. No	Course content	Hrs	% Weightage
1	Pointers <ul style="list-style-type: none"> • Introduction and Features of Pointers • Declaration of Pointer • Initialization of Pointer • Void Pointers • Array of Pointers • Pointers to Pointers • Pointer to Functions 	11	20%
2	Structure and Union Structure: <ul style="list-style-type: none"> • Introduction,Features • Declaration and Initialization and accessing • Array of structures • typedef • Enumerated data type Union: <ul style="list-style-type: none"> • Introduction • Declaration and Initialization and accessing 	12	30%
3	File Handling <ul style="list-style-type: none"> • Definition of file • Streams and File Types • Opening , Reading , closing of File (Text Mode) • Reading From and writing into Files (Text mode) 	11	25%
4	Dynamic Memory Allocation <ul style="list-style-type: none"> • Introduction • Dynamic Memory Allocation • Allocating a Block of Memory: Malloc • Allocating Multiple Blocks of Memory: Calloc • Releasing the used Space: Free • Altering the size of a Block:Realloc 	11	25%

Continuous Assessment:

Sr. No.	Active Learning Activities	Marks
1	Real-World Application Analysis: Students will Analyze and make report on real-world applications developed in C and upload it on GMIU Web Portal.	10
2	Bug Detectives :	10



	Students will be assigned a code containing intentional errors, students need to identify errors, correct and upload on GMIU Web Portal.	
3	Skill-Building Task: Mini project definition will be given; Students have to upload the task on GMIU Web Portal.(group of four)	10
4	Code Refactoring: Provide students with code snippets and challenge them to refactor for better readability and performance and upload on GMIU Web Portal.	10
5	Attendance	10
Total		50

Suggested Specification table with Marks (Theory):75

Distribution of Theory Marks (Revised Bloom's Taxonomy)						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage	25%	45%	15%	15%	0	0

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcome:

After learning the course the students should be able to:	
CO1	Develop, debug and execute programs to perform memory access using Pointers.
CO2	Implement different data types under a single structure and Utilize memory effectively using Union.
CO3	Implement file and I/O operations in C.
CO4	Understand dynamic memory management techniques.
CO5	Write and implement C programs to solve various problems using advance C programming features.

List of Practical

Sr. No	Descriptions	Unit No	Hrs
1	Develop a C program to declare a pointer and initialize it to point to a variable.	1	2
2	Develop a C program that demonstrates the use of pointers by swapping two integers using a pointer.	1	2



3	Develop a C Program to demonstrates the use of Array of pointers.	1	2
4	Develop a C Program to demonstrates the use of pointers to pointers.	1	2
5	Develop a C program to use of pointer to Functions.	2	2
6	Develop a C program that demonstrates the use of a structure.	2	2
7	Develop C program that demonstrates the use of structures to store and display information about students.	2	2
8	Develop a C program using Array of structures.	2	2
9	Write a C Program to demonstrate usage of enum and typedef.	2	2
10	Develop a C Program to demonstrate the use of union.	2	2
11	Develop a C Programs using file operations. Opening , Reading , closing of File (Text Mode).	3	2
12	Develop a C Programs using file operations. Reading From and writing into Files (Text mode).	3	2
13	Develop a C Program to demonstrate use of malloc, calloc.	4	2
14	Develop a C Program to Find the largest element using Dynamic Memory Allocation.	4	2
15	Develop a C program to demonstrate use of free, realloc.	4	2
		Total	30

Instructional Method:

The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

From the content 10% topics are suggested for flipped mode instruction.

Students will use supplementary resources such as online videos, NPTEL/SWAYAM videos, e-courses, Virtual Laboratory.

The internal evaluation will be done on the basis of Active Learning Assignment.

Practical/Viva examination will be conducted at the end of semester for evaluation of performance of students in laboratory.

Reference Books:

- [1] Programming in ANSI C : E. Balagurusamy
- [2] Programming with ANSI and Turbo C: Ashok N. Kamthane
- [3] Let us 'C' : Yashavant Kanetkar
- [4] C Programming for Problem Solving : Dr. Guruprasad Nagraj

