GAINS
GANDAKI ACADEMY
OF INTERDISCIPLINARY STUDIES
EMPOWERING FUTURES

BSc-CSIT
FAR WESTERN UNIVERSITY

Course Title:	**Programming Fundamentals and 'C' Programming**	**Credit:**	**3**
Course Code:	**CSIT 115**	**Number of periods per week:**	**3**
Nature of Course:	**Theory+Lab**	**Total Hours:**	**48**
Year:	**First**	**Semester:**	**First**

1. **Course Introduction**
   The course intends to enable the students to be acquainted with the basic concepts of programming methodology, 'C' Programming language.
2. **Objectives**
   At the end of this course, the students should be able to:
   - To develop a programming logic.
   - To teach basic principles of programming.
   - To develop skills for writing programs using 'C'.

3. Specific Objectives and Contents

Unit 1: Introduction To Algorithms and C– 8 LH	
Specific Objectives • Define algorithm, use of algorithms • Describe different notations of algorithms • State standard notations and common functions • Classify different Pseudo-code Conventions • Develop fundamental algorithms • Write different algorithms for different problems • Differentiate different programming approaches and their benefits. • Understand the basic structure of C program • Understand different types of data types and qualifiers in terms of memory requirement and range. • Write various programs using different data types, and qualifiers.	**Contents** **Fundamentals of algorithms:** Notion of an algorithm. Pseudo-code conventions like assignment statements and basic control structures. **Algorithmic problems:** Develop fundamental algorithms for (i)Exchange the values of two variables with and without temporary variable, (ii) Counting positive numbers from a set of integers, (iii) Summation of set of numbers, (iv) Reversing the digits of an integer, (v) Find smallest positive divisor of an integer other then 1, (vi) Find G.C.D. and L.C.M. of two as well as three positive integers (vii) Generating prime numbers. **Different approaches in programming:** Procedural approach, Object Oriented approach, Event Driven approach. **Structure of C:** Header and body, Use of comments, Compilation of program. **Data Concepts:** Variables, Constants, data types like: int, float char, double and void. Qualifiers: Short and ling size qualifiers, signed and unsigned qualifiers. Declaring variables. cope of the variables according to block. Hierarchy of data types.
Unit 2: Basic of C – 4 LH	
Specific Objectives • Write various 'C' programs to perform • various types of operations on the data values which are to be processed. • Input various types of data and obtain the output in a desired form • Alter the sequence of the execution of the program • Set up loops to repeat a set of statements, desired number of times	**Contents** **Types of operators:** Arithmetic, Relational, Logical, Compound Assignment, Increment and decrement, Conditional or ternary, Bitwise and Comma operators, Precedence and order of evaluation. Statements and Expressions. **Type Conversions:** Automatic and Explicit type conversion **Data Input and Output function:**

• transfer control to different statements in the program	Formatted I/O: printf(), scanf(), Character I/O format : getch(), gerche(), getchar(), getc(), gets(),putchar(), putc(), puts() **Iterations:** Control statements for decision making: (i) Branching: if statement, else.. If statement, switch statement (ii) Looping: while loop, do… while, for loop. (iii) Jump statements: break, continue and goto.
Unit 3: Arrays, Strings and Sorting Techniques- 8 LH	
Specific Objectives • Understand what arrays are • What is the need for arrays • How arrays can be used in C Language • Declare and use one-dimensional and two-dimensional arrays • Understand the need for character and string variables • Declare and use character and string variables • Use functions to handle character and string data • Understand the Purpose of Sorting • Understand the different methods of Sorting. • Identify the advantages of different algorithms of Sorting • Be able to write programs in C to implement the algorithms for Sorting • Explain what is meant by Efficiency of an algorithm • Compare algorithms for Efficiency	**Contents** **Arrays:** (One and multidimensional), declaring array variables, initialization of arrays, and accessing array elements. **Strings:** Declaring and initializing String variables. Character and string handling functions. **Sorting Algorithms:** Bubble, Selection,
Unit 4: Functions, Storage Classes and Recursion- 8 LH	
Specific Objectives • Understand what Functions are and why are they needed. • Be able to define a Function in terms of its arguments and return values • Understand when and how to use Functions • Understand what are Macros and why they are needed • Explain how Macros are different from functions? • Understand what is Recursion? • Explain the Advantages of Recursion • Write programs for some standard situations for recursive functions such as Fibonacci Sequence and Towers of Hanoi • Be able to understand situations where recursion is needed • Understand the concept of a storage class	**Contents** **Functions:** Global and local variables, Function definition, return statement, Calling a function by value, Macros in C, Different between functions and macros. **Storage classes:** Automatic variables, External variables, Static variables, Register variables. **Recursion:** Definition, Recursion function algorithms for factorial, Fibonacci sequence, Tower of Hanoi. Implement using C

• Understand the different storage classes • Understand the concept of scope, visibility and longevity of a variable • Understand which storage class should be used under what circumstances • Learn the advantages and disadvantages of each storage class	
Unit 5: Structure and Union- 4 LH	
Specific Objectives • Understand what are structures and why they are needed • Be able to define a structure • Be able to read and assign values to elements in a structure • Be able to understand the relationship between arrays and structures • Be able to define structures within structures • Be able to understand the relationship between structures and functions • Be able to understand what are unions • Write programs involving the use of structures	**Contents** **Structure:** Declaration of structure, reading and assignment of structure variables, Array of structures, arrays within structures, within structures, structures and functions. **Unions:** Defining and working with union
Unit 6: Pointers and File Handling- 6 LH	
Specific Objectives • Understand the pointers • Write dynamic programs • Understand strength of pointers • Store data in files • Read data from files • Understand File Handling Functions	**Content** **Fundamentals:** Fundamentals, Pointer variables, Referencing and dereferencing, Pointer Arithmetic, Chain of pointers, Pointers and Arrays, Pointers and Strings, Array of Pointers, Pointers as function arguments, Functions returning pointers, Pointer to function, Pointer to structure, Pointers within structure. **File Handling:** Different types of files like text and binary, Different types of functions fopen(), fclose(), fputc(), fscanf(), fprintf(), getw(), putw(), fread(), fwrite(), fseek() **Dynamic Memory Allocation:** Malloc(), calloc(), realloc(), free() and size of operator.
Unit 7: Link Lists- 4 LH	
Specific Objectives • Define a Linear Link List and list its features. • Understand the advantages & shortcomings of link list over an array. • Differentiate between Link List & Array. • Write & Explain the basic operations of Linear Link List. • Understand how to implement a link list. • Write a program in C to implement linear link list	**Content** **Linear Link lists:** Representation of link list in memory, Algorithms for traversing a link list, searching a particular node in link list, insertion into link list (insertion at the beginning of a node, insertion after a given node) deletion from a link list. Implement using C.

Unit 8: Stacks- 3 LH	
Specific Objectives • Define a stack and its features. • Write Algorithms for the basic operations of Stack. • Understand the difference between Stack & Array. • Understand how an Array is used to implement a Stack. • Write a program in C to implement Stack.	**Content** **Stacks:** Definition, Array representation of stacks, Algorithms for basic operators to add and delete an element from the stack, Implement using C.
Unit 9: Queues -3 LH	
Specific Objectives • Define a queue and state its features. • State the applications that use queues. • State the basic operations of a queue. • Differentiate between straight queue and circular queue. • Implement queues using arrays and linked lists.	**Content** **Queues:** Representation of queue, Algorithm for insertion and deletion of an element in a queue, Implement using C.

3. Evaluation System

Undergraduate Program				
External Evaluation	**Marks**	**Internal Evaluation**	**Weightage**	**Marks**
Semester End Examination	60	Assignments	10%	40
		Quizzes	10%	
		Attendance	10%	
		Presentation	10%	
		Term Papers	10%	
		Mid-Term Examination	40%	
		Group Work	10%	
Total External	60	Total Internal	100%	40
Full Mark: 60 + 40				100

4. External Evaluation

End semester examination: It is a written examination at the end of the semester. The questions will be asked covering all the units of the course. The question model, full marks, time and others will be as per the following grid.

Nature of Question	Total Questions to be asked	Total Questions to be Answered	Total Marks	Weight
Group A: Very short answer questions	8	8	8 X 3 = 24	24%
Group B: Short answer type questions	6	5	5 X 8 = 40	40%
Group C: Long answer type questions/case studies	4	3	3 X 12 = 36	36%
			100	**100%**

Each student must secure at least 50% marks in internal evaluation in order to appear in the end semester examination. Failing to get such score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear in the end semester examinations.

Practical examination: Practical examination will be taken at the end of the semester. Students must demonstrate the knowledge of the subject matter.

Internal evaluation:

Assignment:
Each student must submit the assignment individually. The stipulated time for submission of the assignment will be seriously taken.

Quizzes:
Unannounced and announced quizzes/tests will be taken by the respective subject teachers. Such quizzes/tests will be conducted twice per semester. The students will be evaluated accordingly.

Attendance in class:
Students should regularly attend and participate in class discussion. Eighty percent class attendance is mandatory for the students to enable them to appear in the end semester examination. Below 80% attendance in the class will signify NOT QUALIFIED (NQ) to attend the end semester examination.

Presentation:
Students will be divided into groups and each group will be provided with a topic for presentation. It will be evaluated individually as well as group-wise. Individual students have to make presentations on the given topics.

Term paper:
A term paper must be prepared using a computer in a standard technical writing format and contain the required number of pages. It should be prepared and submitted individually. The stipulated time for submission of the paper will be taken seriously as one of the major criteria of the evaluation.

Mid-term examination:
It is a written examination and the questions will be asked covering all the topics in the course session.

Discussion and participation:
Students will be evaluated based on their active participation in the classroom discussions.

Instructional Techniques:
All topics are discussed with emphasis on real-world application. The list of instructional techniques is as follows:

- Lecture and Discussion

- Group work and Individual work
- Self-study
- Assignments
- Presentation by Students
- Term Paper writing

Recommended Books

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (n.d.). *Introduction to algorithms* (2nd ed.). PHI. (For chapter 1, 2, 3, 10)

Lipschutz, S. (n.d.). *Data structures (Schaum's outline series in computers*). McGraw-Hill Book Company. (For chapter 2, 5, 6, 9)

Balagurusamy, E. (n.d.). *Programming in ANSI C (3rd ed.)*. TMH. (For chapter 2 to 13)

Knuth, D. E. (n.d.). *The art of computer programming, volume 1*: Fundamental algorithms. Narosa Publishing House.

Loudon, K. (n.d.). *Mastering algorithms with* C. Shroff Publishers.

Sedgewick, R. (n.d.). *Algorithms in C (3rd ed.)*. Pearson Education Asia.

Gilberg, R. F., & Forouzan, B. A. (n.d.). *Data structures: A pseudocode approach with* C. Thomas.

Kanetkar, Y. (n.d.). *Let us* C. BPB.

Kumar, R., & Agrawal, R. (n.d.). *Programming in ANSI* C. TMH.

Gottfried, B. S., & Chhabra, J. K. (n.d.). *Programming with* C (2nd ed.). Schaum's Outlines (TMH).

Venugopal, K. R., & Prasad, S. R. (n.d.). *Programming with* C. TMH Outline Series.

Bhave, M. D., & Patekar, S. A. (n.d.). *Unix and* C. Nandu Printer and Publishers Private Limited.

Course Title:	**Programming Fundamentals and C Programming**	**Credit:**	**1**
Course Code:	**CSIT 115**	**Number of periods per week:**	**1**
Nature of Course:	**Lab**	**Total Hours:**	**16**
Year:	**First**	**Semester:**	**First**

Objectives:

By the end of the course the student should be able to conceptualize C and apply it in real work settings

Guidelines for the lab work:

1. There should be a lab book for the practical work related to the subject
2. The lab book will contain details of all practical's to be conducted in the lab
3. Students should read the lab book before coming to the lab
4. Every practical should have:
a. Title
b. Objectives c. Description d. Examples
e. Self-Activities
i. Objective questions
ii. Sample programs to be typed and executed f. Task list to be decided by the lab in-charge.
g. Outputs to be verified by the lab in-charge.
5. Each practical should be conducted in the following manner:
a. Explanation by lab in-charge – 10 minutes b. Self-activities by students
c. Lab in-charge will allocate tasks to each student (selection from a list / modify given task / specify new task)
d. At the end of the slot, the lab in-charge has to verify the outputs and give a remark (Complete / Incomplete / Needs Improvement)

Evaluation
The practical exam will be graded based on the following marking scheme:

- In-semester evaluation (Lab Book or Journal) 20 %
- Final Exam Written 60 %
- Final Exam Oral 20 %

Assignment List for Lab Work

Assignment List for Lab Work

All the students will have to complete the following set of programming. Lab in-charge may assign

additional assignment depending upon the time available.

1. Assignment to demonstrate use of data types, simple operators (expressions)
2. Assignment to demonstrate decision making statements (if and if-else, nested structures)
3. Assignment to demonstrate decision making statements (switch case)
4. Assignment to demonstrate use of simple loops
5. Assignment to demonstrate use of nested loops
6. Assignment to demonstrate menu driven programs.
7. Assignment to demonstrate writing C programs in modular way (use of user defined functions)

8. Assignment to demonstrate recursive functions.
9. Assignment to demonstrate use of arrays (1-d arrays) and functions
10. Assignment to demonstrate use of multidimensional array (2-d arrays) and functions
11. Assignment to demonstrate use of pointers
12. Assignment to demonstrate concept of strings (string & pointers)
13. Assignment to demonstrate array of strings.
14. Assignment to demonstrate use of bitwise operators.
15. Assignment to demonstrate structures (using array and functions)
16. Assignment to demonstrate structures and unions
17. Assignment to demonstrate command line arguments and pre-processor directives.
18. Assignment to demonstrate file handling (text files)
19. Assignment to demonstrate file handling (binary files and random access to files)
20. Assignment to demonstrate graphics using C

Prescribed Books

- Deitel, P. J., & Deitel, H. M. (n.d.). *How to program* (2nd ed.). Pearson Education.
- Kelley, A., & Pohl, I. (n.d.). *A book on* C. Pearson Education.
- Kernighan, B. W., & Ritchie, D. M. (n.d.). *The C programming language*. PHI.
- Gottfried, B. S. (n.d.). *Programming with* C. TMH.
- Kochan, S. G. (n.d.). *Programming in* C. CBS Publishers & Distributors.
- Kanetkar, Y. (n.d.). *Let us* C. BPB Publications.
- Schildt, H. (n.d.). Complete C reference.
- Forouzan, B. A., & Gilberg, R. F. (n.d.). *Structured programming approach using* C. Thomson Learning Publications.