Pokhara University
Faculty of Science and Technology

Course No.: CMP 124	Full marks: 100
Course title: **Programming in C (3-1-3)**	Pass marks: 45
Nature of the course: Theory & Practical	Total Lectures: 45 hrs
Level: Bachelor	Program: BE (Computer, IT and Software)

1. Course Description

This course is designed to develop the skills in students to use the C language, which follows the structured programming paradigm, to develop the computer programs. It introduces the different generations of programming languages, the origin, strengths and basic constructs of the C language. After completion of this course, the students will be able to use the C language to resolve a given problem through the problem solving steps- problem analysis, design of algorithm and flowchart, coding using the C language, executing and compiling the developed program, testing and debugging the program and finally well documenting the program for the future understanding.

2. General Objectives

- To acquaint the students with basic knowledge of computer language and generations of programming languages.
- To develop the skills in students to solve a given problem using computer program.
- To acquaint the students with the knowledge of structured programming paradigm (using the C language) to develop the computer programs.

3. Methods of Instruction

Lecture, Discussion, Readings, Practical works and Project works.

4. Contents in Detail.

Specific Objectives	**Contents**

- Explain, compare and classify programming languages and its generations. - Explain and compare the types of software. - Design and develop algorithms and flowcharts. - Familiarize with computer program documentation.	**Unit 1: Programming languages and problem solving (6 hrs)** 1.1 Programming Languages (machine-level language, assembly language and high-level language) and its generations. 1.2 Software and its types 1.3 Structured programming 1.4 Problem solving using computer- Problems Analysis (understanding of the problem, feasibility and requirement analysis), Design (Algorithm and flowchart), Coding (compilation/interpretation and execution), Testing and debugging, Implementation, Evaluation and Maintenance of computer programs, Program documentation
- Describe the origin of the C language and know its strength. - Understand the basic constructs of the C language.	**Unit 2: Introduction to C (6 hrs)** 2.1 History of C language 2.2 Features of C 2.3 The C as a middle-level language 2.4 The C as a system programming language 2.5 The C character set 2.6 Keywords and Identifiers 2.7 Data types 2.8 Constants, variables and their declaration 2.9 Formatted input/output functions 2.10 The C Operators
- Understand and implement the various control structures of the C language.	**Unit 3: Control Structures (6 hrs)** 3.1 Introduction and types of control statements- sequential, branching and looping statements 3.2 Branching statements- simple if statement, if-else, nested if, if-else-if ladder and switch statements 3.3 Looping statements- for loop, while loop, do-while loop, nested loop 3.4 The break, continue and goto statements

- Develop the C program that uses the various types of single and multi-dimensional arrays.	**Unit 4: Arrays and Strings (6 hrs)** 4.1 Introduction to arrays 4.2 One dimensional and Multidimensional arrays 4.3 Initialization of arrays and accessing the elements of arrays 4.4 Strings- the character arrays 4.5 Functions related to the strings
- Design and develop the C programs using functions.	**Unit 5: Functions (6 hrs)** 5.1 Introduction 5.2 Importance of functions 5.3 Returning a value from a function and sending a value to a function 5.4 Function prototypes 5.5 Calling a function- Call by value and Call by reference 5.6 Recursive functions 5.7 Passing an array to a function 5.8 Local variables, formal parameters and global variables 5.9 Storage classes 5.10 Pre-processor directives- C libraries, macros and header files
- Use the pointers in arrays, functions and programs to dynamically allocate and deallocate memory.	**Unit 6: Pointers (6 hrs)** 6.1 Introduction 6.2 Pointer operators 6.3 Pointer arithmetic 6.4 Returning multiple values form functions using pointers 6.5 Pointers and Arrays 6.6 Double indirection 6.7 Dynamic memory allocation

- Use the structures and unions to store and access the heterogenous data required in a program.	**Unit 7: Structure and Union (5 hrs)** 7.1 Definition of Structure 7.2 Nested-Structure 7.3 Array of Structure 7.4 Structures and Pointers 7.5 Union 7.6 Self-referential structure
- Use the C file handling concepts to store the data permanently in a computer file and access them whenever required.	**Unit 8: Files and File Handling (4 hrs)** 8.1 FILE pointer, File opening modes (read, write, append) 8.2 File handling functions 8.3 Creating and operating a file in different modes

5. **Practical Works**

Laboratory works of 45 hours per group of maximum 24 students should cover all the concepts of C language studied in the lectures. Students should submit a final project that uses all the constructs and features of C studied in this course. The marks for the practical work will be based on the project work.

6. List of Tutorials:

The various tutorial activities that suits this course should cover all the content of this course to give student a space to engage more actively with the course content in the presence of instructor. Students should submit tutorials as assignments or class-works to the instructor for evaluation. The following tutorial activities of 15 hours per group of maximum 24 students should be conducted to cover the content of this course:

A. Discussion-based Tutorials: (2 hrs)

1. Evolution of Programming languages and its generations (Class discussion)
2. Software and its types.
3. Generations of programming languages.
4. Structured programming. (Oral Presentation).

B. Problem solving-based Tutorials: (10 hrs)

1. Develop algorithms and flowcharts to solve various problems such as to find largest number among three numbers, prime numbers, temperature conversion, product of matrices, finding sum of the terms in series, printing various patterns etc.
2. Develop the C programs for the problems for which you developed the algorithms.
3. Write a program to pass an array to a function.
4. Write a program to use pointers to pass multiple values from a function.
5. Write a program to use the basic string functions to manipulate string data.

6. Write a program to use the principle of recursion to solve the complex problems such as to find factorial of a number, fibonacci series.
7. Write a program to illustrate the macros and header files.
8. Write a program to to illustrate how memory is allocated and deallocated in C language.
9. Write a program to use the nested structure. Discuss the scenarios when the structures and unions are used in real practice.
10. Write a program to solve simple file handing problems.

C. Review and Question/Answer-based Tutorials: (3 hrs)

1. Case study of "Development of C with the UNIX operating system and origin of C++ languages" followed by Oral Presentation in class.
2. Students ask questions within from the course content and assignments and review key course content in preparation for tests or exams.

7. Evaluation system and Students' Responsibilities

Internal Evaluation

The internal evaluation of a student may consist of assignments, attendance, internal assessment, lab reports and project works etc. The internal evaluation scheme for this course is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		30	Semester-End examination	50
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Practical		20		
Attendance & Class Participation	10%			
Lab Report/Project Report	20%			
Practical Exam/Project Work	40%			
Viva	30%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Student Responsibilities:

Each student must secure at least 45% marks separately in internal assessment and practical evaluation with 80% attendance in the class in order to appear in the Semester End Examination. Failing to get such score will be given NOT QUALIFIED (NQ) to appear the Semester-End Examinations. Students are advised to attend all the classes, formal exam, test, etc. and complete all the assignments within the specified time period. Students are required to complete all the requirements defined for the completion of the course.

8. Prescribed Books and References

Text Books:

1. Balagurusamy, E. (2016). *Programming In Ansi C.* Tata McGraw-Hill.
2. Gottfried, Byron S. (2018). *Programming with C.* Tata McGraw-Hill.

References:

1. Kelley A. & Pohl I.(2001). *A Book on C, Programming in C.* Addison-Wesley.
2. Kernighan, B. W., & Ritchie, D. M. (2002). *The C programming language.* Prentice Hall.