Course Title: Programming Concept with C
Course No. : ICT. Ed. 412   Nature of course: Theoretical + Practical
Level: B.Ed.   Credit hours: 3 (2T+1P)
Semester: First   Teaching hours: 80 (32T+48P)

1.   **Course Description**
      The aim of the course is to impart knowledge of the basic concepts of procedural programming
      and to help the students build skills for solving problems using procedural programme. It provides
      the students with the basic features of the language such as data types, operators, control structure,
      array, functions, structure, pointer and file handling which are the common features of any
      programming languages. Students are more engaged in laboratory work to exaction of programing
      experiments rather than theoretical concept.

2.   **General Objectives of the Course**
      Following are the general objective of this course:
      - To make the student knowledgeable about the procedural programming concept.
      - To enable the student in implement the essential programming concepts and methods in
        practices.
      - To explore the programming execution procedure compiler, memory and library.
      - To provide the students with the skills of application to solve the real world problems.

3.   **Specific Objectives and Contents**

Specific Objectives	Contents

- Give an introduction of programming language - Describe assembler, compiler and interpreter - State syntax and semantics - Explain programming design tools - Indentify the features of good programme - Design algorithm and draw diagram of flow chart of sequence, decision making and repetition concept of programming	**Unit 1: Introduction to Programming Concept** (6)
	  1.1 Introduction of Programming Language
	  1.2 Assembler, Compiler and Interpreter
	  1.3 Syntax and Semantics
	  1.4 Programming Design Tools
	      1.4.1 Algorithm
	      1.4.2 Flow chart
	      1.4.3 Pseudo codes
	  1.5 Features of good programme
	Lab Works
	  - Designing algorithm and draw flow chart for sequence, decision making and repetition concept of general programming.

- Outline historical development of C programming language. - Describe basic structure of C programme, character set, token and comments, variables and constants. - Apply data types and conversion in programme. - Outline input and output in relation to C - Apply operators and operands in programme.	**Unit 2: Introduction to C** (10)
	  2.1 History of C Programme
	  2.2 Basic Structure of C Programme
	  2.3 Character set, Token and Comments
	  2.4 Variables and Constants
	  2.5 Data Types
	  2.6 Type Conversion
	  2.7 Input and Output
	  2.8 Operators
	      2.8.1 Arithmetic
	      2.8.2 Relational
	      2.8.3 Logical

	2.8.4 Increment/Decrement 2.8.5 Assignment 2.8.6 Bitwise **Lab Works** • Developing basic structure of C programme. • Declaring and assigning variables and constants. • Applying input and output build in function • Using arithmetic operators • Giving demo of type conversion
• Create different selective structure programme. • Apply loop concept in programme. • Describe the interrupt concept in programme.	**Unit 3: Control Structure** (14) **3.1** Selective Structure 3.1.1 If statement 3.1.2 If-else statement 3.1.3 Nested if-else statement 3.1.4 Switch statement 3.1.5 Conditional operator (:?) **3.2** Looping structure: 3.2.1 While Loop 3.2.2 Do-while loop, 3.2.3 For loop 3.2.4 Nested Loops 3.2.5 Loop interrupts **Practical Works** • User if, if-else and switch statement • Use while, do-while, for and nested loop concept.
• Clarify the concept of functions. • Create function with different parameter passing methods. • Apply recursion in programme.	**Unit 4: Function** (10) 4.1 Function Concept 4.2 Function prototype, call and definition 4.3 Different ways of using function 4.4 Call by value, call by reference 4.5 Recursion **Practical Works** • Prototype, call and define function • Pass the different parameter methods • Use call by reference methods using function. • Create a recursion function.
• Generate the programme with use of array. • Demonstrate pointer and address references. • Use string function to write program. • Apply pointer in array and string function.	**Unit 5: Array, Pointer and String** (10) 5.1 Concept of array 5.2 Array declare, access and initialization. 5.3 Multi-dimensional array 5.4 Concept of Pointer 5.5 Pointer address, dereference, declaration, assignment, initialization 5.6 Pointer Arithmetic 5.7 Array and Pointer 5.8 String 5.9 String functions in C 5.10 Pointer and String

	Practical Works • Array declares, define, initialize. • Creating a single or multi-dimensional array. • Using pointer and demo for arithmetic function. • Using different string function in program.
• Create a programme to heterogeneous data types using array. • Apply the pointer in address referencing mode.	**Unit 6: Structure and Union** (8) 6.1 Concept of Structure 6.2 Initializing, accessing member of structure 6.3 Array of structure 6.4 Pointer to structure 6.5 Union 6.6 Different between union and structure **Practical Works** • Creating structure data types with application of loop. • Creating union data types.
• Define the concept the file handling in C. • To apply file access methods. • Apply to input and out formatting structures. • Apply error handling methods.	**Unit 7: Input output and File Handling** (12) 7.1 Concept of File handling 7.2 File Access methods 7.3 Functions of file handling: fopen(), fclose(), fflush(), freopen() 7.4 Formatted input out 7.5 Character input output 7.6 Direct input output 7.7 Random file access 7.8 Error handling 7.9 File operation **Practical Works** • Creating file handling application for open, read, write and appends. • Handling the random access files. • Applying the text formatting function.
• Create application to using input, output and storage concept using C	**Unit 8: Project** (10) Preparation of simple application using C with the feature of input, process, output and store in external file.

Note: The figures in parenthesis indicate approximate teaching hours allotted to respective units.

1.1 General Instructional Techniques
Lecture preferably with the use of multi-media projector, demonstration, practical classes, discussion, and brain storming are used in all units.

4.2 Specific Instructional Techniques

Demonstration is an essential instructional technique for all units in this course during teaching-learning process. Specifically, demonstration with practical works will be specific instructional technique in this course. The details of suggested instructional techniques are presented below:

Units	Activities

Unit 1: For drawing different types of algorithm, flowchart and pseudo codes.	• Demonstration by the teacher on drawing different types of algorithm, flowchart and pseudo codes • Individual lab work by each student • Monitoring of students' work by reaching each student and providing feedback for improvement • Presentation by students followed by peers' comments and teacher's feedback
Unit 2 to 7:	• Demonstration by the teacher on project works mentioned in each unit • Lab work individually or in pairs in tasks assigned by the teacher • Monitoring of students' work by reaching each individual or pair and providing feedback for improvement • Presentation by students followed by peers' comments and teacher's feedback
Unit 8:	• Demonstration by the teacher on simple application with input process and store in notepad using C • Lab work individually or in pairs in tasks assigned by the teacher • Monitoring of students' work by reaching each individual or pair and providing feedback for improvement • Presentation by students followed by peers' comments and teacher's

5. Evaluation

Evaluation of students' performance is divided into parts: Internal assessment and internal and external practical examination and theoretical examinations. The distribution of points is given below:

Internal Assessment	External Practical Exam/Viva	Semester Examination (Theoretical exam)	Total Points
40 Points	20 Points	40 Points	100 Points

Note: Students must pass separately in internal assessment, external practical exam and semester examination.

5.1 Internal Assessment (40 Points)

Internal assessment will be conducted by subject teacher based on following criteria:

1) Class Attendance   5 points
2) Learning activities and class performance   5 points
3) First assignment (written assignment)   10 points
4) Second assignment (Case Study/project work with presentation)   10 points
5) Terminal Examination   10 Points

        Total   40 points

5.2   Semester Examination (40 Points)

Examination Division, Dean office will conduct final examination at the end of semester.

Objective question (Multiple choice questions 10 x 1 point)   10 Points
Short answer questions (6 questions x 5 marks)   30 Points

        Total   40 points

5.3 Practical Exam/Viva (20 Points)

Examination Division, Dean Office will conduct final practical examination at the end of semester. Practical record book, practical written test, demonstration of practical activities and viva are assessment indicators.

6. Recommended Books and References materials (including relevant published articles in national and international journals)

Kanetkar, Y. P. (2008). *Let us C* 8th Ed, New Delhi, BPB Publication (Unit 1 -8)

Baral, D.S., Baral, D. & Ghimire S. K, The secretes of C programming language. Kathmandu,

Bhundi Purana Prakasan. (Unit 1-8)

Balagurusamy, E. (2007). Programming in ANSI C. New Delhi, India: Tata McGraw-Hill.

References materials:

Raman, R. (1984). Computer programming in C, New Delhi, PHI. India (unit 1-8)

B.S. Gottfried(2001), Schaum's Outline Series for Programming with C, Second Edition, Tata

McGraw Hill Publishing Company, New Delhi

David Griffiths (2012). Head First C. Shroff