C Programming
EG2101CT

Year: II **Total: 7 hours /week**
Part: I **Lecture: 4 hours/week**
Tutorial: hour/week
Practical: hours/week
Lab: 3 hours/week

Course description:
This course deals with the problem-solving techniques using C programming language. It provides the students with the knowledge of the basic features of the C language such as data types, keywords, operators, control structure, array, String handling functions, functions, structure and union, pointer and file handling.

Course objectives:
After completion of this course students will be able to:
1. Implement fundamentals concepts of programming language.
2. Apply sequential, conditional and looping statements while developing programs.
3. Create modular programs using array.
4. Make and apply programs using function, strings, string handling function, structure and union, pointer and data files.

Course Contents:

Theory

Unit 1. Programming Language Fundamentals [6 Hrs.]
- 1.1. Introduction to Program and Programming Language
- 1.2. Types of Programming Language (Low Level and High-Level Language)
- 1.3. Language Translator (Assembler, Compiler and Interpreter)
- 1.4. Program Error, Types of Error (Syntax, Semantic, Runtime Error)
- 1.5. Program Design Tools (Algorithm, Flowchart)

Unit 2. Introduction to C [8 Hrs.]
- 2.1. Overview and History of C
- 2.2. Features, Advantages and Disadvantages of C
- 2.3. Structure of C Program, Compiling Process
- 2.4. Character set used in C, Data types, Variables. C Tokens (Keywords, Identifier, Constants, Operators), Header files, Library function
- 2.5. Preprocessor Directives, Escape Sequence, Comments
- 2.6. Input Output Operation
  - 2.6.1. Formatted input/output function (printf(), scanf())
  - 2.6.2. Unformatted input/output function (getchar(), putchar(), gets(), puts(), getc(), putc())

Unit 3. Operators and Expressions [4 Hrs.]
- 3.1. Operators, Operand, Operation, Expression
- 3.2. Types of Operators (Unary, Binary, Ternary, Arithmetic, Relational, Logical, Assignment, Increment/Decrement, Conditional, Bitwise, Size-of Operators)

Unit 4. Control Structure/Statement [12 Hrs.]
- 4.1. Sequential Statement
- 4.2. Decision/Selection/Conditional Statement
  - 4.2.1. if statement

4.2.2. if…else statement
4.2.3. if…else if…else statement
4.2.4. Nested if…else statement
4.2.5. Switch statement
4.3. Loop (for, while and do-while)
4.4. Jump statement (break, continue, goto statement)

Unit 5. Array and String **[8 Hrs.]**

5.1. Introduction to Array, Declaration, Initialization
5.2. Types of Arrays (1-D Array, Multi-dimensional Array)
5.3. String, Array of String
5.4. String Handling Function (strlen(), strrev(), strupr(), strlwr(), strcpy(), strcat(), strcmp())

Unit 6. Function **[6 Hrs.]**

6.1. Introduction
6.2. Function components (function declaration, function call, function definition)
6.3. Types of function (library/built-in function and user-defined function)
6.4. Category of function according to return value and arguments
6.5. Parameter passing in C (call by value and call by reference)
6.6. Recursion (recursive function)
6.7. Passing array to function
6.8. Passing string to function

Unit 7. Structure and Union **[6 Hrs.]**

7.1. Structure: definition, declaration, initialization, size of structure
7.2. Accessing member of Structure
7.3. Array of Structure
7.4. Nested Structure
7.5. Union: definition, declaration, size of union
7.6. Structure Vs. Union

Unit 8. Pointer **[4 Hrs.]**

8.1. Introduction to Pointer
8.2. Address (&) and indirection (*) operator
8.3. Pointer Arithmetic Operations
8.4. Pointer to Pointer in C
8.5. Dynamic Memory Allocation (malloc(), calloc(), realloc(), free())

Unit 9. Data files **[6 Hrs.]**

9.1. Introduction to data files
9.2. Types of files (text file, binary file)
9.3. File handling operation
9.4. Opening and closing file
9.5. Creating file
9.6. Library functions for READING from a file and WRITING to a file: (fputs, fgets, fputc, fgetc fprintf, fscanf)

Practical: **[45 Hrs.]**

1. Write programs to implement sequential structure.

2. Write programs to implement conditional structure.
3. Write programs to implement looping structure.
4. Write programs to implement array and string handling function.
5. Write programs to implement library function, user defined function and recursive function.
6. Write programs to implement structure and union.
7. Write programs to implement pointer.
8. Write programs to read from a file and write to data file using fputs, fgets, fputc, fgetc fprintf, fscanf function.

Final written exam evaluation scheme			
Unit	**Title**	**Hours**	**Marks Distribution***
1	Programming Language Fundamentals	6	8
2	Introduction to C	8	11
3	Operators and Expressions	4	5
4	Control Structure/Statement	12	16
5	Array and String	8	11
6	Function	6	8
7	Structure and Union	6	8
8	Pointer	4	5
9	Data files	6	8
	Total	**60**	**80**

* There may be minor deviation in marks distribution.

References:

1. Gotterfried, B. (2001). Programming with C. (3rd ed.). India: Mcgraw Hill Education.
2. Bhatta, R.D. (2015). A Text Book of C Programming. (3rd ed.). Nepal: Vidyarthi Pustak Bhandar.
3. Thareja, R. (2015). Introduction to C Programming. (2nd ed.). India: Oxford University Press.
4. Kantekar, Y. (2012). Let us C. (10th ed.). India: BPB Publications.
5. Balagurusamy, E. (2008). Programming in ANSI C. (6th ed.). India: The McGraw Hill Companies.