

## CONCEPTS OF COMPUTER & 'C' PROGRAMMING (ECS-101/102)

**Teacher Name:**

**Ms. Urvashi Saraswat**

### Course Structure

Sr. No.	Course Type	Course Code	Course Name	Credits	Details of Sessional Marks				ESM	Total Marks
					CT	TA	Lab	Total		
1	ESC	ECS-101 / ECS-102	Computer Concepts & 'C' Programming	4 (3-0-2)	15	20	15	50	50	100

**Prerequisite: NIL**

### Course Content:

#### Unit-1:

Introduction to Computers: Computer hardware Components, peripherals and their functions, Number Systems and conversion methods, Concept of an algorithm; termination and correctness. Algorithms to programs: specification, top-down development and stepwise refinement, Introduction to programming environment, use of high level programming language for the systematic development of programs. Introduction to the design and implementation of correct, efficient and maintainable programs, Structured Programming, Trace an algorithm to depict the logic.

#### Unit-2:

Basic operating System Concepts: Introduction of MS-DOS, WINDOWS, and LINUX Operating Systems, Functional Knowledge of these operating systems, Introduction of basic commands of LINUX and Editors, Managing Files and Directories in LINUX, Programming Environment in LINUX, Writing and executing programs in LINUX.

#### Unit-3:

Programming in C: History, Introduction to C Programming Languages, Structure of C programs, compilation and execution of C programs, Debugging Techniques, Data Types and Sizes, Declaration of variables, Modifiers, Identifiers and keywords, Symbolic constants,

Storage classes (automatic, external, register and static), Enumerations, command line parameters, Macros, The C Preprocessor.

#### **Unit-4:**

Operators: Unary operators, Arithmetic & logical operators, Bit wise operators, Assignment operators and expressions, Conditional expressions, Precedence and order of evaluation. Control statements: if-else, switch, break, and continue, the comma operator, goto statement. Loops: for, while, do-while. Functions: built-in and user-defined, function declaration, definition and function call, and parameter passing: call by value, call by reference, recursive functions, Multi-file programs. Arrays: linear arrays, multidimensional arrays, passing arrays to functions, Arrays and strings.

#### **Unit-5:**

Structure and Union: definition and differences, self-referential structure. Pointers: value at (\*) and address of (&) operator, pointer to pointer, Dynamic Memory Allocation, calloc and malloc functions, array of pointers, function of pointers, structures and pointers. File Handling in C: opening and closing a data file, creating a data file, read and write functions, unformatted data files.

#### **Lab Work:**

1. Write C program to find largest of three integers.
2. Write C program to check whether the given string is palindrome or not.
3. Write C program to find whether the given integer is
  - (i). a prime number
  - (ii). an Armstrong number.
4. Write C program for Pascal triangle.
5. Write C program to find sum and average of n integer using linear array.
6. Write C program to perform addition, multiplication, transpose on matrices.
7. Write C program to find Fibonacci series of iterative method using user-defined function.
8. Write C program to find factorial of n by recursion using user-defined functions.
9. Write C program to perform following operations by using user defined functions:
  - (i) Concatenation
  - (ii) Reverse
  - (iii) String Matching
10. Write C program to find sum of n terms of series:  $n - n*2/2! + n*3/3! - n*4/4! + \dots$
11. Write C program to interchange two values using
  - (i). Call by value.
  - (ii). Call by reference.
12. Write C program to sort the list of integers using dynamic memory allocation.
13. Write C program to display the mark sheet of a student using structure.
14. Write C program to perform following operations on data files:
  - (i) Read from data file.
  - (ii) Write to data file.

15. Write C program to copy the content of one file to another file using command line argument.

### **Text and References Books:**

1. Kernighan, Ritchie, "The C Programming Language", PHI
2. V. Rajaraman, "Fundamentals of Computers", PHI
3. Peter Norton's, "Introduction to Computers", TMH
4. Gottfried, "Programming in C", Schaum's Series, Tata McGraw Hill
5. Yashwant Kanitkar, "Working with C", BPB
6. E. Balagurusamy, "Programming in ANSI C", TMH

### **Course Outcomes:**

1. Identify the parts of the computer system and explain the functioning of its components alongwith the process of problem solving. (Remember, Understand)
2. Design an algorithmic solution for a given problem and translate it into a program. (Design)
3. Understand different operating systems, related concepts and their functions. (Understand)
4. Use the appropriate control statements to solve the given problem. (Apply)
5. Implement different Operations on arrays and use functions to solve the given problem. (Apply)
6. Understand pointers, structures and unions & Implement file Operations in C programming. (Understand, Apply)

## **Lecture Plan - Session 2018-19**

**Department: Computer Science & Engineering**  
**Concepts of Computer Programming in 'C'- CS-1<sup>st</sup> yr**

<b>S.No.</b>	<b>Name of Topic</b>	<b>No. of lectures taken</b>
1.	Introduction To Computers: Computer hardware Components, peripherals and their functions	1
2.	Number Systems and conversion methods	2
3.	Concept of an algorithm; termination and correctness	1
4.	Algorithms to programs: specification, top-down development and stepwise refinement	1
5.	Introduction to Programming Environment	1
6.	Use of high-level programming language for the systematic development of programs	1
7.	Introduction to the design and implementation of correct, efficient and maintainable programs	1
8.	Structured Programming, Trace an algorithm to depict the logic.	1
9.	Basic operating System Concepts: Introduction of MS-DOS, WINDOWS, and LINUX Operating Systems, Functional Knowledge of these operating systems	2
10.	Introduction of Basic Commands of LINUX and Editors	2
11.	Managing Files and Directories in LINUX, Programming Environment in LINUX, Writing and executing programs in LINUX	2
12.	Programming in C: History, Introduction to C Programming Languages, Structure of C programs, compilation and execution of C programs	1
13.	Debugging Techniques, Data Types and Sizes	1
14.	Declaration of variables, Modifiers, Identifiers and keywords, Symbolic constants, Storage classes (automatic, external, register and static), Enumerations, command line parameters	3
15.	Macros, The C Preprocessor	1
16.	Operators: Unary operators, Arithmetic & logical operators, Bit wise operators, Assignment operators and expressions, Conditional expressions, Precedence and order of evaluation.	2
17.	Control statements: if-else, switch, break, and continue, the comma operator, goto statement.	3
18.	Loops: for, while, do-while	2
19.	Functions: built-in and user-defined, function declaration, definition and function call, and parameter passing: call by value, call by reference, recursive functions, Multi file programs.	3
20.	Arrays: linear arrays, multidimensional arrays, passing arrays to functions, Arrays and strings.	3
21.	Structure and Union: definition and differences, self-referential structure.	2
22.	Pointers: value at (*) and address of (&) operator, pointer to pointer, Dynamic Memory Allocation, calloc and malloc functions, array of pointers, function of pointers, structures and pointers.	3
23.	File Handling in C: opening and closing a data file, creating a data file, read and write functions, unformatted data files.	2

## Assignments

### Assignment-1

Q1. Define “What do you understand by A Computer”. Give a well-defined block diagram, representing all blocks of a computer system.

Q2. Explain: Bits, Bytes and Word.

Q3. How does memory and storage differ. Explain the role of RAM and ROM in a computer memory.

Q4. Give explanation of “working an industry with”:

- a. Mainframes
- b. Supercomputers
- c. Minicomputers
- d. Microcomputers
- e. Smartphones and embedded computers

Q5. What is the role of processing unit. And how does CPU handles different operations.

### ASSIGNMENT-2

Q1. Give the correctness of algorithm. Explain with the help of an example.

Q2. What are:

- a. High-level languages
- b. Assembler
- c. Compiler
- d. Interpreter

Q3. What do you understand by Top-down development of an algorithm?

Q4. What are advantages and disadvantages of Structured Programming?

Q5. Draw a flowchart for the following :

- a) to find greater and smaller number from given two numbers.
- b) to calculate sum of first 10 odd numbers.

Q6. Convert the following numbers as:-

- a.  $(11110.01101)_2 \rightarrow (?)_{10}$
- b.  $(357.46)_{10} \rightarrow (?)_2$
- c.  $(110101)_2 \rightarrow (?)_8$
- d.  $(456)_8 \rightarrow (?)_2$
- e.  $(A1B)_{16} \rightarrow (?)_2$

### ASSIGNMENT-3

- Q1. Differentiate among MS-DOS, WINDOWS and LINUX operating systems.
- Q2. What is base operating system of LINUX and how does LINUX differs with it?
- Q3. Give the fundamentals of writing and executing programs in LINUX.
- Q4. Give any five commands of MS-DOS and LINUX, along their role and syntax of operation.
- Q5. How are files and directories managed in LINUX?
- Q6. Display all prime numbers between 50 and 150.
- Q7. Write a program to swap the values of two numbers. Do this using call by reference method of function.
- Q8. Write a program to accept 10 values in an integer array. Display the number of odd, even, and negative numbers.

### ASSIGNMENT-4

- Q1. Give a structure of C program.
- Q2. What is the size and range of:  
a. Integer                      b. Double                      c. Float                      d. Short int                      e. Long int
- Q3. What is the role of: modifier, integer and storage class. Give examples.
- Q4. Explain the role of "Macros" in C.
- Q5. Write a program:  
a. To swap two numbers (without using third variable).  
b. To add all the digits of a number.  
c. To reverse a number.
- Q6. If ip is a pointer to an integer, what does ip++ mean? What does  
`*ip++ = 0;` do?
- Q7. How much memory does the call `malloc(10)` allocate? What if you want enough memory for 10 ints?
- Q8. Write a C program `cmi.c` that accepts a distance in inches and prints the corresponding value in cms. Note that 1 inch = 2.54 cm. *Test data and expected output:*

Enter the distance in inches:3

Distance 3.00 inches is = 7.62 cms

Q9. Create a structure to store the employee number, name, department and basic salary. Create a array of structure to accept and display the values of 10 employees.

Q10. Following is the menu to be displayed to the user. On selecting a choice display appropriate result. Number should be accepted from the user.

Menu

1. Prime Factors
2. Leap Year
3. Sum of all digits
4. Number in reverse order

### **ASSIGNMENT-5**

Q1. Write a program to generate a calculator with: addition, subtraction, multiplication and division, using:

- a. If-else
- b. switch

Q2. Write a program to sort an array in ascending order. The array should have 50 elements and the input shall be taken from the user.

Q3. Write a program to display following patterns::

A. \*

```
  **
 ***
****
*****
```

B. 1

```
  11
 111
1111
11111
```

C. 1

```
  12
 123
1234
12345
```

D. 54321

```
 54321
 5432
 543
 54
 5
```

E.           \*

              \* \*

              \* \* \*

              \* \* \* \*

**Q4.** The assignment in

```
char c;  
int *ip = &c;                   /* WRONG */
```

is in error; you can't mix char pointers and int pointers like this. How, then, is possible to write

```
char *cp = malloc(10);  
int *ip = malloc(sizeof(int));
```

 without error on either line?

**Q5.** Perform the following operations on integer array of 10 elements. Accept the values from user.

1. Sort an array in ascending order.
2. Display sum of all odd values stored in an array.
3. Display number of even values stored in an array.