

VIJAYANAGARA SHREEKRISHNADEVARAYA UNIVERSITY
JNANASAGARA CAMPUS, BALLARI-583105

B.Sc., Computer Science

Course Syllabus



With effect from the academic year 2021-22 onwards

Department of Studies in Computer Science

Semester : I

Course Code: DSC1

Course Title: Computer Fundamentals and Programming in C

Course Credits: 04

Hour of Teaching/Week: 04

Total Contact Hours: 52

Formative Assessment Marks: 40

Exam Marks: 60

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Confidently operate Desktop Computers to carry out computational tasks
- Understand working of Hardware and Software and the importance of operating systems
- Understand programming languages, number systems, peripheral devices, networking, multimedia and internet concepts
- Read, understand and trace the execution of programs written in C language
- Write the C code for a given problem
- Perform input and output operations using programs in C
- Write programs that perform operations on arrays

Unit – 1

Fundamentals of Computers: Introduction to Computers - Computer Definition, Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organisation of a Digital Computer; Number Systems – different types, conversion from one number system to another; Computer Codes – BCD, Gray Code, ASCII and Unicode; Boolean Algebra – Boolean Operators with Truth Tables; Types of Software – System Software and Utility Software; Computer Languages - Machine Level, Assembly Level & High Level Languages, Translator Programs – Assembler, Interpreter and Compiler; Planning a Computer Program - Algorithm, Flowchart and Pseudo code with Examples. **8 hrs**

Unit – 2

Introduction to C Programming: Over View of C; History and Features of C; Structure of a C Program with Examples; Creating and Executing a C Program; Compilation process in C. **10 hrs**

C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants.

Input and output with C: Formatted I/O functions - printf and scanf, control strings and escape sequences, output specifications with printf functions; Unformatted I/O functions to read and display single character and a string - getchar, putchar, gets and puts functions.

Unit – 3

C Operators & Expressions: Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associativity; Evaluation of arithmetic **12 hrs**

Expressions; Type conversion.

Control Structures: Decision making Statements - Simple if, if_else, nested if_else, else_if ladder, Switch-case, goto, break & continue statements; Looping Statements - Entry controlled and Exit controlled statements, while, do-while, for loops, Nested loops.

Unit - 4

Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; **12 hrs**

Two Dimensional arrays - Declaration, Initialization and Memory representation.

Strings: Declaring & Initializing string variables; String handling functions -

strlen, strcmp, strcpy and strcat; Character handling functions - toascii, toupper, tolower, isalpha, isnumeric etc.

Pointers in C: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Advantages and disadvantages of using pointers;

Unit - 5

User Defined Functions: Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type. **10 hrs**

User defined data types: Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.

Text Books:

1. Pradeep K. Sinha and Priti Sinha: Computer Fundamentals (Sixth Edition), BPB Publication
2. E. Balgurusamy: Programming in ANSI C (TMH)

References:

1. Kamthane: Programming with ANSI and TURBO C (Pearson Education)
2. V. Rajaraman: Programming in C (PHI - EEE)
3. S. ByronGottfried: Programming with C (TMH)
4. Kernighan & Ritchie: The C Programming Language (PHI)
5. Yashwant Kanitkar: Let us C
6. P.B. Kottur: Programming in C (Sapna Book House)

Course Code: DSC1

Course Credits: 02

Total Contact Hours: 52

Formative Assessment Marks: 25

Course Title: C Programming Lab

Hour of Teaching/Week: 04

Exam Marks: 25

Practice Lab

The following activities be carried out/ discussed in the lab during the initial period of the semester.

1. Basic Computer Proficiency
 - a. Familiarization of Computer Hardware Parts
 - b. Basic Computer Operations and Maintenance.
 - c. Do's and Don'ts, Safety Guidelines in Computer Lab
2. Familiarization of Basic Software – Operating System, Word Processors, Internet Browsers, Integrated Development Environment (IDE) with Examples.
3. Type Program Code, Debug and Compile basic programs covering C Programming fundamentals discussed during theory classes.

Programming

Lab Part A:

1. Write a C Program to read radius of a circle and to find area and circumference
2. Write a C Program to read three numbers and find the biggest of three
3. Write a C Program to demonstrate library functions in *math.h*
4. Write a C Program to check for prime
5. Write a C Program to generate n primes
6. Write a C Program to read a number, find the sum of the digits, reverse the number and check it for palindrome
7. Write a C Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers
8. Write a C Program to read percentage of marks and to display appropriate message (Demonstration of else-if ladder)
9. Write a C Program to find the roots of quadratic equation (demonstration of switch-case statement)
10. Write a C program to read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
11. Write a C Program to remove Duplicate Element in a single dimensional Array
12. Program to perform addition and subtraction of Matrices

Part B:

1. Write a C Program to find the length of a string without using built in function
2. Write a C Program to demonstrate string functions.
3. Write a C Program to demonstrate pointers in C
4. Write a C Program to check a number for prime by defining *isprime()* function
5. Write a C Program to read, display and to find the trace of a square matrix
6. Write a C Program to read, display and add two m x n matrices using functions
7. Write a C Program to read, display and multiply two m x n matrices using functions
8. Write a C Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
9. Write a C Program to Reverse a String using Pointer
10. Write a C Program to Swap Two Numbers using Pointers
11. Write a C Program to demonstrate student structure to read & display records of n students.
12. Write a C Program to demonstrate the difference between structure & union.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course

Open Elective courses offered by the Department of Computer Science

Course Code: OEC1

Course Credits: 03

Total Contact Hours: 42

Formative Assessment Marks: 40

Course Title: C Programming Concepts

Hour of Teaching/Week: 03

SEE: 60

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Confidently operate Desktop Computers to carry out computational tasks
- Understand working of Hardware and Software and the importance of operating systems
- Understand programming languages, number systems, peripheral devices, networking, multimedia and internet concepts
- Read, understand and trace the execution of programs written in C language
- Write the C code for a given problem
- Perform input and output operations using programs in C
- Write programs that perform operations on arrays

Unit – 1

Fundamentals of Computers: Introduction to Computers - Computer Definition, **8 hrs** Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organisation of a Digital Computer; Number Systems – different types, conversion from one number system to another; Computer Codes – BCD, Gray Code, ASCII and Unicode; Boolean Algebra – Boolean Operators with Truth Tables; Types of Software – System Software and Utility Software; Computer Languages - Machine Level, Assembly Level & High Level Languages, Translator Programs – Assembler, Interpreter and Compiler; Planning a Computer Program - Algorithm, Flowchart and Pseudo code with Examples.

Unit – 2

Introduction to C Programming: Over View of C; History and Features of C; Structure **10 hrs** of a C Program with Examples; Creating and Executing a C Program; Compilation process in C.

C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants.

Input and output with C: Formatted I/O functions - printf and scanf, control strings and escape sequences, output specifications with printf functions; Unformatted I/O functions to read and display single character and a string - getchar, putchar, gets and puts functions.

Unit – 3

C Operators & Expressions: Arithmetic operators; Relational operators; Logical **8 hrs**

operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associativity; Evaluation of arithmetic expressions; Type conversion.

Control Structures: Decision making Statements - Simple if, if_else, nested if_else, else_if ladder, Switch-case, goto, break & continue statements; Looping Statements - Entry controlled and Exit controlled statements, while, do-while, for loops, Nested loops.

Unit - 4

Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; **8 hrs**
Two Dimensional arrays - Declaration, Initialization and Memory representation.

Strings: Declaring & Initializing string variables; String handling functions - strlen, strcmp, strcpy and strcat; Character handling functions - toascii, toupper, tolower, isalpha, isnumeric etc.

Unit - 5

User Defined Functions: Need for user defined functions; Format of C user defined **8 hrs** functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type.

Text Books

1. Pradeep K. Sinha and Priti Sinha: Computer Fundamentals (Sixth Edition), BPB Publication
2. E. Balgurusamy: Programming in ANSI C (TMH)

References

1. Kamthane: Programming with ANSI and TURBO C (Pearson Education)
2. V. Rajaraman: Programming in C (PHI - EEE)
3. S. Byron Gottfried: Programming with C (TMH)
4. Kernighan & Ritchie: The C Programming Language (PHI)
5. Yashwant Kanitkar: Let us C
6. P.B. Kottur: Programming in C (Sapna Book House)