

COURSE-WISE SYLLABUS**Semester I**

Year	I	Course Code:21BSC1C1CS1L		Credits	04
Sem.	1	Course Title: Computer Fundamentals and Programming in C		Hours	52
Course Pre-requisites, if any		NA			
Formative Assessment Marks: 40		Summative Assessment Marks: 60		Duration of ESA:.02 hrs.	
Course Outcomes	After completing this course satisfactorily, a student will be able to: <ul style="list-style-type: none">• 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 No.	Course Content			Hours	
Unit I	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 andPseudo code with Examples. 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.			13	
Unit II	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 stings and escapesequences, output specifications with printf functions; Unformatted I/O functions to read and display single character and a string - getchar, putchar, gets and			13	

	puts functions. 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 expressions; Type conversion.	
Unit III	<p>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.</p> <p>Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays - Declaration, Initialization and Memory representation.</p> <p>Strings: Declaring & Initializing string variables; String handling functions - <i>strlen</i>, <i>strcmp</i>, <i>strcpy</i> and <i>strcat</i>; Character handling functions - <i>tolower</i>, <i>toupper</i>, <i>isalpha</i>, <i>isnumeric</i> etc.</p>	13
Unit IV	<p>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;</p> <p>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.</p> <p>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.</p>	13
Recommended Learning Resources		
Print Resources	<p>Text Books</p> <ol style="list-style-type: none"> 1. Pradeep K. Sinha and Priti Sinha: Computer Fundamentals (Sixth Edition), BPB Publication 2. E. Balgurusamy: Programming in ANSI C (TMH) <p>References</p> <ol style="list-style-type: none"> 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 	

Year	I	Course Code: 21BSC1C1CS1P		Credits	02
Sem.	I	Course Title:C Programming Lab		Hours	45
Course Pre-requisites, if any:		Knowledge of Programming			
Formative Assessment Marks: 25		Summative Assessment Marks: 25	Duration of ESA: 03 hrs.		
<u>Practice Labs</u> 1. 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.					
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 <i>math.h</i> 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 dimensionalArray 12. Program to perform addition and subtraction ofMatrices					
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 <i>isprime()</i> 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

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Activity – 1 from Part A	Write up on the activity/ task	3
	Demonstration of the activity/ task	07
Activity-2 from Part B	Write up on the activity/ task	3
	Demonstration of the activity/ task	07
Viva based on Lab Activities		05
Total		25

OPEN-ELECTIVE SYLLABUS :

Year	I	Course Code: 21BSC1O1CS1			Credits	03
Sem.	1	Course Title: C Programming Concepts			Hours	40
Course Pre-requisites, if any		NA				
Formative Assessment Marks: 40		Summative Assessment Marks: 60		Duration of ESA:.02 hrs.		
Course Outcomes	At the end of the course the student should be able to: 1. Read, understand and trace the execution of programs written in C language 2. Write the C code for a given problem 3. Perform input and output operations using programs in C 4. Write programs that perform operations on arrays 5. Write user defined functions to perform a task					
Unit No.	Course Content					Hours
Unit I	Introduction to C Programming: Overview of C; History and Features of C; Structure 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.					10
Unit II	Input and output with C: Formatted I/O functions – <i>printf</i> and <i>scanf</i> , control strings and escape sequences, output specifications with <i>printf</i> functions; Unformatted I/O functions to read and display single character and a string - <i>getchar</i> , <i>putchar</i> , <i>gets</i> and <i>puts</i> functions 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 expressions; Type conversion. Control Structures: Decision making Statements - <i>Simple if</i> , <i>if_else</i> , <i>nested if_else</i> , <i>else_if ladder</i> , <i>Switch Case</i> , <i>goto</i> , <i>break</i> & <i>continue</i> statements					10
Unit III	Looping Statements - Entry controlled and exit controlled statements, <i>while</i> , <i>do-while</i> , <i>for</i> loops, Nested loops. Derived data types in C: Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays - Declaration, Initialization and Memory representation. Strings: Declaring & Initializing string variables; String handling functions - <i>strlen</i> , <i>strcmp</i> , <i>strcpy</i> and <i>strcat</i> ; Character handling functions - <i>toascii</i> , <i>toupper</i> , <i>tolower</i> , <i>isalpha</i> , <i>isnumeric</i> etc					10

Unit IV	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
Recommended Learning Resources		
Print Resources	<p>Text Books:</p> <ol style="list-style-type: none"> 1. C: The Complete Reference, By Herbert Schildt. 2. C Programming Language, By Brian W. Kernighan 3. Kernighan & Ritchie: The C Programming Language (PHI) <p>References</p> <ol style="list-style-type: none"> 1. E. Balaguruswamy: Programming in ANSI C (TMH) 2. Kamthane: Programming with ANSI and TURBO C (Pearson Education) 3. V. Rajaraman: Programming in C (PHI –EEE) 4. S. Byron Gottfried: Programming with C (TMH) 5. Yashwant Kanitkar: Let us C 	