JSS COLLEGE OF ARTS, COMMERCE AND SCIENCE

(Autonomous)

Ooty Road, Mysuru-570025

Model Curriculum Structures for
Bachelor of Science (Basic and Honours) Programme
with Computer Science as Major & Minor Course
Model Syllabus for I and II Semesters
and
Open Elective Courses in Computer Science

As per

NATIONAL EDUCATION POLICY - 2020
(NEP-2020)

2021-22

DEPARTMENT OF COMPUTER SCIENCE

The objectives of the B.Sc. - Computer Science Program

1. The primary objective of this program is to provide a foundation of computing principles for effectively using information systems and enterprise softwares.
2. It helps students analyze the requirements for system programming and exposes students for information systems
3. This programme provides students with options to specialize in various software system.
4. To produce outstanding Computer Scientists who can apply the theoretical knowledge into practice in the real world and develop standalone live projects themselves
5. To provide opportunity for the study of modern methods of information processing and its applications.
6. To develop among students the programming techniques and the problem- solving skills through programming
7. To prepare students who wish to go on to further studies in computer science and related subjects.
8. To acquaint students to Work effectively with a range of current, standard, Office Productivity software applications.

Program Outcomes:

1. **Discipline knowledge:** Acquiring knowledge on basics of Computer Science and ability to apply to design principles in the development of solutions for problems of varying complexity
2. **Problem Solving:** Improved reasoning with strong mathematical ability to Identify, formulate and analyze problems related to computer science and exhibiting a sound knowledge on data structures and algorithms.
3. **Design and Development of Solutions:** Ability to design and developmentof algorithmic solutions to real world problems.
4. **Programming a computer:** Exhibiting strong skills required to program a computer for various issues and problems of day-to-day scientific applications.
5. **Application Systems Knowledge**: Possessing a minimum knowledge to practice existing computer application software.
6. **Communication:** Must have a reasonably good communication knowledge both in oral and writing.
7. **Ethics on Profession, Environment and Society:** Exhibiting professional ethics to maintain the integrality in a working environment and also have concern on societal impacts due to computer-based solutions for problems.
8. **Lifelong Learning:** Should become an independent learner. So, learn to learn ability.
9. **Motivation to take up Higher Studies:** Inspiration to continue educations towards advanced studies on Computer Science.

Additional Program Outcomes for B. Sc. (Hons) in Computer Science

The four years Bachelors in Computer Science (Hons) program enables students to attain the following additional attributes besides the afore-mentioned attributes:

1. Apply standard Software Engineering practices and strategies in real -time software project development
2. Design and develop computer programs/computer-based systems in the areas related to AI, algorithms, networking, web design, cloud computing, IoT and data analytics.
3. Acquaint with the contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems
4. The ability to apply the knowledge and understanding noted above to the analysis of a given information handling problem.
5. The ability to work independently on a substantial software project and as an effective team member.

IIIA. Model Program Structures for the Under-Graduate Programs in Universities and Colleges in Karnataka

Bachelor of Science (Basic/Hons.) in subjects with practical with both subjects as majors

Sem.	Discipline Core (DSC) (Credits) (L+T+P)	Discipline Elective (DSE) / Open Elective (OE) (Credits) (L+T+P)	Ability Enhancement Compulsory Courses (AECC),		Skill Enhancement Courses (SEC)			Total Credits
					Skill based credits (L+T+P)	Value based (Credits) (L+T+P)		
I	Discipline A1(4+2) Discipline B1(4+2)	OE-1 (3)	L1-1(3), L2-1(3) (4 hrs. each)	Environmental Studies (3)		Physical Education for fitness (1) (0+0+2)	Health & Wellness (1) (0+0+2)	25
II	Discipline A2(4+2) Discipline B2(4+2)	OE-2 (3)	L1-2(3), L2-2(3) (4 hrs. each)		SEC-1: Digital Fluency (2)(1+0+2)	Physical Education - Yoga(1) (0+0+2)	NCC/NSS/R&R(S&G) /Cultural (1) (0+0+2)	25
Exit option with Certificate (50 credits)								
III	Discipline A3(4+2) Discipline B3(4+2)	OE-3 (3)	L1-3(3), L2-3(3) (4 hrs. each)	Constitution of India (3)		Physical Education- Sports (1) (0+0+2)	NCC/NSS/R&R(S&G) /Cultural (1)(0+0+2)	25
IV	Discipline A4(4+2) Discipline B4(4+2)	OE-4 (3)	L1-4(3), L2-4(3) (4 hrs. each)		SEC-2: Artificial Intelligence (2) (1+0+2)	Physical Education -Games (1) (0+0+2)	NCC/NSS/R&R(S&G) /Cultural (1)(0+0+2)	25
Exit option with Diploma (100 credits) or continue the third year with both the subjects as majors								
V	Discipline A5(3+2), Discipline A6(3+2) Discipline B5(3+2), Discipline B6(3+2)				SEC-3: SEC such as Cyber Security (2) (1+0+2)			23
VI	Discipline A7(3+2), Discipline A8(3+2) Discipline B7(3+2) Discipline B8(3+2) Internship (2)				SEC-4: Professional Communication (2)			24
Exit option with Bachelor of Arts, B.A./ Bachelor of Science, B. Sc. Basic Degree (146 credits) or choose one of the Disciplines as Major								
VII	Discipline A/B-9(3+2) Discipline A/B-10(3+2) Discipline A/B-11(3)		DS-A/B Elective-1(3) DS-A/B Elective-2(3) Res.Methodology(3)					22
VIII	Discipline A/B-12(3+2) Discipline A/B-13(3) Discipline A/B-14(3)		DS-A/B Elective-3(3) Research Project (6) *					20

NEP 2020 Syllabus – B.Sc. (Cs M) and B.Sc. (P Cs) 2021-22 onwards

Year	**Sem**	**Combin ation**	**Course Code**	**Title**	**Hours / Week**			**Credits**			**Maximum Marks**						**Exam Duration**	**Total Marks**
											Th. IA		**Pr. IA**		**Exam**			
					L	**T**	**P**	**L**	**T**	**P**	**C1**	**C2**	**C1**	**C2**	**Th.**	**Pr.**		
I	I	**Cs M**	**FSA45034** [DSC-1]	Computer Fundamentals and Programming in C	**4**	**0**	**0**	**4**	**0**	**0**	**20**	**20**	**-**	**-**	**60**	**-**	**3 Hours**	**100**
		P Cs	**FSA45035** [DSC-1]	Computer Fundamentals and Programming in C	**4**	**0**	**0**	**4**	**0**	**0**	**20**	**20**	**-**	**-**	**60**	**-**	**3 Hours**	**100**
		Cs M	**FSA45034** [DSC-1L]	C Programming Lab	**0**	**0**	**4**	**0**	**0**	**2**	**-**	**-**	**10**	**15***	**-**	**25**	**3 Hours**	**50**
		P Cs	**FSA45035** [DSC-1L]	C Programming Lab	**0**	**0**	**4**	**0**	**0**	**2**	**-**	**-**	**10**	**15***	**-**	**25**	**3 Hours**	**50**
		IBA EG	FHA21031	SEC-DIGITAL FLUENCY	**1**	**0**	**1**	**1**	**0**	**2**	**-**	**-**	**10**	**15***	**-**	**25**	**1 Hours**	**50**
		IBA HP	FHA21032		**1**	**0**	**1**	**1**	**0**	**2**	**-**	**-**	**10**	**15***	**-**	**25**	**1 Hours**	**50**
		IBA KG	FHA21033		**1**	**0**	**1**	**1**	**0**	**2**	**-**	**-**	**10**	**15***	**-**	**25**	**1 Hours**	**50**
		IBA HE	FHA21034		**1**	**0**	**1**	**1**	**0**	**2**	**-**	**-**	**10**	**15***	**-**	**25**	**1 Hours**	**50**
		IBA JP	FHA21035		**1**	**0**	**1**	**1**	**0**	**2**	**-**	**-**	**10**	**15***	**-**	**25**	**1 Hours**	**50**
			FSA880	**OE-Open Elective Office Automation**	**3**	**0**	**0**	**3**	**0**	**0**	**-**	**-**	**10**	**15***	**-**	**25**	**2 Hours**	**50**
			FSA890	**OE- C – Programming Concepts**	**3**	**0**	**0**	**3**	**0**	**0**	**-**	**-**	**10**	**15***	**-**	**25**	**2 Hours**	**50**
	II	**Cs M**	**FSB45034** [DSC-2]	Data Structures using C	**4**	**0**	**0**	**4**	**0**	**0**	**20**	**20**	**-**	**-**	**60**	**-**	**3 Hours**	**100**
		P Cs	**FSB45035** [DSC-2]	Data Structures using C	**4**	**0**	**0**	**4**	**0**	**0**	**20**	**20**	**-**	**-**	**60**	**-**	**3 Hours**	**100**
		Cs M	**FSB45034** [DSC-2 L]	Data structures Lab	**0**	**0**	**4**	**0**	**0**	**2**	**-**	**-**	**10**	**15***	**-**	**25**	**3 Hours**	**50**

		P Cs	**FSB45035** [DSC-2 L]	**Data structures Lab**	**0**	**0**	**4**	**0**	**0**	**2**	**-**	**-**	**10**	**15***	**-**	**25**	**3 Hours**	**50**
		I BCA	**FAB210**	**SEC-DIGITAL FLUENCY**														
		I B.Sc. PC	**FSB21031**															
		I B.Sc. PM	**FSB21032**															
		I B.Sc. PE	**FSB21033**															
		I B.Sc. CsM	**FSB21034**															
		I B.Sc. PCs	**FSB21035**															
		I B.Sc. BtZ	**FSB21036**															
		I B.Sc. CBt	**FSB21037**															
		I B.Sc. CZ	**FSB21038**															
		I B.Sc. BZ	**FSB21039**															
		I B.Sc. BtBc	**FSB21040**															
		I B.Sc. MbBt	**FSB21041**															
		I B.Sc. MbBc	**FSB21042**															
			FSB880	**OE – E Commerce**	**3**	**0**	**0**	**4**	**0**	**0**	**-**	**-**	**10**	**15***	**-**	**25**	**2 Hours**	**50**
			FSB890	**OE-Web Designing**	**3**	**0**	**0**	**4**	**0**	**0**	**-**	**-**	**10**	**15***	**-**	**25**	**2Hours**	**50**

Note: 15* is spilt 10 marks for Practical's C2 + 5 marks for practical Record/Report

Discipline Specific Elective Courses:		
Group 1: • IoT • Cyber Law and Cyber Security • Web Programming - PHP and MySQL • Clouds, Grids, and Clusters • Software Testing	**Group-2:** • Information and Network Security • Data Compression • Discrete Structures • Opensource Programming • Multimedia Computing • Big Data	**Group-3:** • Data Analytics • Storage Area Networks • Pattern Recognition • Digital Image Processing • Parallel Programming • Digital Signal Processing
Open Electives in Computer Science: (For BA, BSc, BCom, BSW, BBA, BBM students studying Core Courses other than Computer Science/ Computer Applications)		
• Office Automation • Multimedia Processing • Computer Animation • Accounting Package	• C Programming Concepts • Python Programming Concepts • R Programming	• E-Content Development • E-Commerce • Web Designing

Model Syllabus for BSc (Basic and Honors), Semesters I and II

Semester: I

Course Code: DSC-1	**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	Exam Duration: 03

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

Course Content

Content	**Hours**
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. **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 - 2	
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 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. **C Operators & Expressions:** Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associatively; Evaluation of arithmetic expressions; Type conversion.	13
Unit - 3	
Control Structures: Decision making Statements - S*imple 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.	13

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 - *strlen, strcmp, strcpy and strcat;* Character handling functions - *toascii, toupper, tolower, isalpha, isnumeric* etc.	
Unit - 4	
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; **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. **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.	13

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 & Ritche: The C Programming Language (PHI)
5. Yashwant Kanitkar: Let us C
6. P.B. Kottur: Programming in C (Sapna Book House)

Course Code: **DSC-1L**	**Course Title: C Programming Lab**
Course Credits: 02	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 03

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

Evaluation Scheme for Lab Examination

Assessment Criteria		**Marks**
Program – 1 from Part A	Flowchart / Algorithm	02
	Writing the Program	05
	Execution and Formatting	03
Program -2 from Part B	Flowchart/Algorithm	02
	Writing the Program	05
	Execution and Formatting	03
Viva Voice based on C Programming		05
Total		**25**

Semester: II

Course Code: **DSC-2**	**Course Title:** Data Structures using C
Course Credits: 04	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 03 Hours

Course Outcomes (COs):

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

- Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms
- Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs
- Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs
- Demonstrate different methods for traversing trees
- Compare alternative implementations of data structures with respect to performance
- Describe the concept of recursion, give examples of its use
- Discuss the computational efficiency of the principal algorithms for sorting and searching

Course Content

Content	**Hours**
Unit - 1	
Introduction to data structures: Definition; Types of data structures - Primitive & Non-primitive, Linear and Non-linear; Operations on data structures. Algorithm Specification, Performance Analysis, Performance Measurement Recursion: Definition; Types of recursions; Recursion Technique Examples - Fibonacci numbers, GCD, Binomial coefficient $_nC_r$, Towers of Hanoi; Comparison between iterative and recursive functions.	13
Unit - 2	
Arrays: Basic Concepts – Definition, Declaration, Initialisation, Operations on arrays; Types of arrays; Arrays as abstract data types (ADT); Representation of Linear Arrays in memory; Traversing linear arrays; Inserting and deleting elements; Sorting – Selection sort, Bubble sort, Quick sort, Selection sort, Insertion sort; Searching - Sequential Search, Binary search; Iterative and Recursive searching; Multidimensional arrays; Representation of multidimensional arrays; Sparse matrices. Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - *malloc*, *calloc*, *realloc* and *free.*	13
Unit - 3	
Linked list: Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly liked list, Header liked list, Circular linked list; Representation of Linked list in Memory; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation; Garbage collection. Stacks: Basic Concepts – Definition and Representation of stacks; Operations on stacks; Applications of stacks; Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression using stack; Application of stack in function calls.	13

Unit - 4	
Queues: Basic Concepts – Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues; Trees: Definition; Tree terminologies –node, root node, parent node, ancestors of a node, siblings, terminal & non-terminal nodes, degree of a node, level, edge, path, depth; Binary tree: Type of binary trees - strict binary tree, complete binary tree, binary search tree and heap tree; Array representation of binary tree. Traversal of binary tree; *preorder, inorder* and *postorder* traversal; Reconstruction of a binary tree when any two of the traversals are given.	13

Text Books

1. Satraj Sahani: Fundamentals of Data Structures

References

1. Tanenbaum: Data structures using C (Pearson Education)
2. Kamathane: Introduction to Data structures (Pearson Education)
3. Y. Kanitkar: Data Structures Using C (BPB)
4. Kottur: Data Structure Using C
5. Padma Reddy: Data Structure Using C
6. Sudipa Mukherjee: Data Structures using C – 1000 Problems and Solutions (McGraw Hill Education, 2007))

Course Code: **DSC-2Lab**	**Course Title: Data Structures Lab**
Course Credits: 02	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 03

Programming Lab

Part A:

1. Write a C Program to find GCD using recursive function
2. Write a C Program to display Pascal Triangle using binomial function
3. Write a C Program to generate n Fibonacci numbers using recursive function.
4. Write a C Program to implement Towers of Hanoi.
5. Write a C Program to implement dynamic array, find smallest and largest element of the array.
6. Write a C Program to create two files to store even and odd numbers.
7. Write a C Program to create a file to store student records.
8. Write a C Program to read the names of cities and arrange them alphabetically.
9. Write a C Program to sort the given list using selection sort technique.
10. Write a C Program to sort the given list using bubble sort technique.

Part B:

1. Write a C Program to sort the given list using insertion sort technique.
2. Write a C Program to sort the given list using quick sort technique.
3. Write a C Program to sort the given list using merge sort technique.
4. Write a C Program to search an element using linear search technique.
5. Write a C Program to search an element using recursive binary search technique.
6. Write a C Program to implement Stack.
7. Write a C Program to convert an infix expression to postfix.

8. Write a C Program to implement simple queue.
9. Write a C Program to implement linear linked list.
10. Write a C Program to display traversal of a tree.

Evaluation Scheme for Lab Examination

Assessment Criteria		**Marks**
Program – 1 from Part A	Flowchart / Algorithm	02
	Writing the Program	05
	Execution and Formatting	03
Program -2 from Part B	Flowchart/Algorithm	02
	Writing the Program	05
	Execution and Formatting	03
Viva Voice based on C Programming		05
Total		**25**

Open Elective courses offered by the Department of Computer Science

Open Elective 1: Office Automation
Course Outcomes (COs):

- Be familiar various office automation tools.
- Create and format a document**.**
- Create and analyse data using Excel.
- Create and customize a presentation for a specific topic.

Unit-1 **14 Hrs**

Introduction, Block diagram of a computer, Input and output devices, memory and storage devices, Types of software, Introduction to operating system – functions, types of operating system and examples.
Introduction to word processing – creating and saving a document, formatting a document – Line spacing, paragraph, Fonts, inserting symbols, header and footer, shape, Tables, Find and replace, Mail merge, saving a document in different formats.

Unit-3 **14 Hrs**

Introduction to spread sheet – entering different types of data like text, numbers, date, , functions and formulae- different categories of functions, chart-creating and formatting a chart, filter, working with single and multiple work books, cell referencing, printing and previewing a document.

Unit-3 **14 Hrs**

Introduction to presentation tools-creating and viewing a presentation, applying design template, formatting options, inserting different objects in a presentation, customize a presentation, adding audio to a presentation, Slide animation, preview Slide transitions Slide show options, adding effect to presentation.

Reference books

1. Computer Basics with Office Automation- Archana Kumar, Dreamtech press, First Edition.
2. The Handbook of Office Automation- Ralph Tomas Reilly, Iuniverse publication, First Edition.

Open Elective-2: C Programming Concepts

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 **14 Hrs**

Fundamentals of Computers: Introduction to Computers -Hardware, software System software, Application software, Utility software, Operating System; Computer Languages – Machine Level, Assembly Level & High-Level Languages, Translator Programs – Assembler, Interpreter and Compiler; Planning a Computer Program – Algorithm and Flowchart 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. C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants.

Unit-2 **14 Hrs**

Input and output with C: Formatted I/O functions - printf and scanf, control stings 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, C Operators & Expressions: Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associatively; 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-3 **14 Hrs**

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.

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 - strlen, strcmp, strcpy and strcat; Character handling functions - toascii, toupper, tolower, isalpha, isnumeric etc. Basics of Pointers in C: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointer Arithmetic; Advantages and disadvantages of using pointers;

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 & Ritche: The C Programming Language (PHI)
5. Yashwant Kanitkar: Let us C 6. P.B. Kottur: Programming in C (Sapna Book House)

Course Content

Semester: I / II

Course Title: **Digital Fluency** Course Credits: 2

Total Contact Hours: 15 hours of theory and 30 hours of practical's

Duration of ESA:

Formative Assessment Marks: 5**0 marks**

Summative Assessment Marks: 50 marks

Model Syllabus Authors:

Course Outcomes (COs):

At the end of the course the student should be able to:

(Write 3-7 course outcomes. Course outcomes are statements of observable student actions that serve as evidence of knowledge, skills and values acquired in this course)

1. Have an intelligent conversation on the key concepts and applications of Artificial Intelligence (AI), Big Data Analytics (BDA), Internet of Things (IoT), Cloud Computing, and Cybersecurity
2. Develop holistically by learning essential skills such as effective communication, problem-solving, design thinking, and teamwork
3. Build his/her personal brand as an agile and expansive learner – one who is interested in horizontal and vertical growth?

Course Articulation Matrix: Mapping of Course Outcomes (COs) with Program Outcomes (POs)

This mapping needs to be done considering POs of respective programs.

Course Outcomes (COs) /
Program Outcomes (POs)
1 2 3 4 5 6 7 8 9 10 11 12

1. Have an intelligent conversation
on the key concepts and
applications of AI, BDA, IoT,
Cloud Computing, and
Cybersecurity
2. Develop holistically by learning
essential skills such as effective
communication,
problem-solving, design thinking,
and teamwork
3. Build his/her personal brand as an
agile and expansive learner – one
who is interested in horizontal and
vertical growth

Course Articulation Matrix relates course outcomes of course with the corresponding program outcomes whose attainment is attempted in this course. Mark 'X' in the intersection cell if a course outcome addresses a particular program outcome.

Course Content (Digital 101)

Details of topic **Duration**

Module 1: **Emerging Technologies** 05 hours

Overview of Emerging Technologies:
i. Artificial Intelligence, Machine Learning, Deep Learning,
ii. Database Management for Data Science, Big Data Analytics,
iii. Internet of Things (IoT) and Industrial Internet of Things (IIoT)
iv. Cloud computing and its service models & v. Cyber Security and Types of cyber attack

Module 2: Applications of Emerging Technologies 05 hours

Applications of emerging technologies:

i. Artificial Intelligence
ii. Big Data Analytics
iii. Internet of Things
iv. Cloud Computing
v. Cyber Security

Module 3: Building Essential Skills Beyond Technology 05 hours

Importance of the following:

i. Effective Communication Skills
ii. Creative Problem Solving & Critical Thinking
iii. Collaboration and Teamwork Skills
iv. Innovation & Design Thinking
v. Use of tools in enhancing skills

References to learning resources:

1. The learning resources made available for the course titled “Digital 101” on Future Skills Prime Platform of NASSCOM