ದಾವಣಗೆರೆ ವಿಶ್ವವಿದ್ಯಾನಿಲಯ, ದಾವಣಗೆರೆ

DAVANGERE UNIVERSITY, DAVANGERE

PROGRAM /COURSE STRUCTURE AND SYLLABUS
as per the Choice Based Credit System (CBCS)
designed in accordance with
Learning Outcomes-Based Curriculum Framework
(LOCF)
of National Education Policy (NEP) 2020
for

Bachelor of Science in
COMPUTER SCIENCE (B.Sc)

w.e.f.

Academic Year 2021-22 and onwards

ಗಣಕ ವಿಜ್ಞಾನ ವಿಭಾಗ

DEPARTMENT OF STUDIES IN COMPUTER SCIENCE

PREAMBLE

Computer Science degrees are some of the most popular study programmes worldwide. In a nutshell, computer science degrees deal with the theoretical foundations of information and computation, taking a scientific and practical approach to computation and its applications. Computation is defined as any type of calculation or use of computing technology that follows well-defined models. Many problems in science, engineering, health care, business, and other areas can be solved effectively with computers, but finding a solution requires both computer science expertise and knowledge of the particular domain. Computer science has a wide range of specialties. These include Computer Architecture, Software Systems, Graphics, Artificial Intelligence, Mathematical and Statistical Analysis, Data Science, Computational Science, and Software Engineering. Universities and other HEIs introduced programmes of computer science. Information Technology is growing rapidly. Increasing applications of computers in almost all areas of human endeavour has led to vibrant industries with concurrent rapid change in technology. Unlike other basic disciplines, developing core competency in this discipline that can be reasonably stable becomes a challenge.

In India, it was initially introduced at the Master (postgraduate) level as MCA and M.Tech. Later on, engineering programmes such as B.Tech and B.E in Computer Science & Engineering and in Information Technology were introduced in various engineering College/Institutions to cater to the growing demand for trained engineering manpower in IT industries. Parallelly, BCA, BSc and MSc programmes with specialization in Computer Science were introduced to train manpower in this highly demanding area. BSc and BSc (Hons) are aimed at undergraduate level training facilitating multiple career paths. Students so graduated, can take up postgraduate programmes in CS (M.Sc)or MCA leading to research as well as R&D, can be employable at IT industries, or can pursue a teaching profession or can adopt a business management career. BSc and BSc (Hons) aims at laying a strong foundation of computer science at an early stage of the career.

There are several employment opportunities and after successful completion of BSc, Some of the jobs that you can get after this course can be in the field of computer science and programming such as IT Analyst, Software Developer, Network Engineer, System Engineer, Computer Operator, Technical Assistant, Technical Engineer, Technical Writer, etc. The prominence in BSc courses, in outcome-based curriculum framework, help students learn solving problems, accomplishing IT tasks, and expressing creativity, both individually and collaboratively. The proposed framework will help Students learn programming techniques and the syntax of one or more programming languages. All students must, therefore, have

access to a computer with a modern programming language installed. The computer science framework does not lay down a explicit language. The educator and students will decide which modern programming languages students will gain knowledge of. More essentially, students will learn to adapt to changes in programming skills and learn new languages as they are developed.

The present Curriculum Framework for BSc degrees is intended to facilitate the students to achieve the following.

• To build up an indulgent and knowledge of the basic theory of Computer Science and Information Technology with good basis on theory, systems and applications such as algorithms, data structures, data handling, data communication, computation and analysis.

• To develop the capability to use this knowledge to analyse new situations in the application domain including software tiny tools.

• To gain indispensable and state-of-the-art skills to take up industry challenges. The objectives and outcomes are carefully designed to suit to the above-mentioned purpose.

• The ability to synthesize the acquired knowledge, understanding and experience for a better and improved comprehension of the real-life problems.

• To learn skills and tools like mathematics, statistics and electronics to find the solution, interpret the results and make predictions for the future developments.

• To formulate, to model, to design solutions, procedure and to use software tools to solve real world problems and evaluate.

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. **Programming a computer:** Exhibiting strong skills required to program a computer for various issues and problems of day-to-day applications with thorough knowledge on programming languages of various levels.

4. **Application Systems Knowledge:** Possessing a sound knowledge on computer application software and ability to design and develop app for applicative problems.

5. **Communication:** Must have a reasonably good communication knowledge both in oral and writing.

6. **Ethics on Profession, Environment and Society:** Exhibiting professional ethics to maintain the internality in a working environment and also have concern on societal impacts due to computer-based solutions for problems.

7. **Lifelong Learning:** Should become an independent learner. So, learn to learn ability.

8. **Motivation to take up Higher Studies:** Inspiration to continue educations towards advanced studies on Computer Science.

Outcome of this program, the students will be able to:

The Bachelor of Computer Science (BSc (Hons)) program makes possible students to accomplish following additional attributes besides the afore-mentioned attributes, by the time of graduation:

1. The ability to apply skill set in computing with strong programming and mathematics skills, as well as wide ranging skills in project management, effective presentations and teamwork.

2. Run by contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems.

3. The capability to apply the knowledge and understanding noted skills to the analysis of a given information handling problems.

4. The skill to work independently on a generous software project and as an effective team member.

DAVANGERE UNIVERSITY

Shivagangothri Campus, Tolahunase, Davangere -577 007

Proposed Curricular and Credits Structure under Choice Based Credit System [CBCS] of Computer Science Major & One Minor Discipline Scheme for the Four Years Computer Science B.Sc. Undergraduate Honors Programme with effect from 2021-22

				SEMESTER-I							
Category	**Course code**	**Title of the Paper**	**Marks**			**Teaching hours/week**			**Credit**	**Duration of exams (Hrs)**
			IA	**SEE**	**Total**	**L**	**T**	**P**		
L1	21BSC1L1LK1	Kannada	40	60	100	4	-	-	3	3
	21BSC1L1LFK1	Functional Kannada								
L2	21BSC1L2LEN2	English	40	60	100	4	-	-	3	3
	21BSC1L2LHI2	Hindi								
	21BSC1L2LSN2	Sanskrit								
	21BSC1L2LTE2	Telugu								
	21BSC1L2LUR2	Urdu								
DSC1	21BSC1C1CS1L	Computer Fundamentals and Programming in C	40	60	100	4	-	-	4	3
	21BSC1C1CS1P	C Programming Lab	25	25	50	-	-	4	2	3
DSC1	Another Department Code	Another Department Course Title	40	60	100	4	-	-	4	3
			25	25	50	-	-	4	2	3
SEC1	21BSC1SE1CS1	Digital Fluency*	25	25	50	1	-	2	2	2
VBC1	21BSC1V1PE1	Physical Education- Yoga	25	-	25	-	-	2	1	-
VBC2	21BSC1V2HW1	Health & Wellness	25	-	25	-	-	2	1	-
OEC1	21BSC1O1CS1	C Programming Concepts	40	60	100	3	-	-	3	3
		Total Marks	**700**			**Semester Credits**			**25**	

Category	Course code	Title of the Paper	IA	SEE	Total	L	T	P	Credit	Duration of exams (Hrs)
SEMESTER-II										
			Marks			**Teaching hours/week**			**Credit**	**Duration of exams (Hrs)**
L3	21BSC2L3LK2	Kannada	40	60	100	4	-	-	3	3
	21BSC2L3FKL2	Functional Kannada								
L4	21BSC2L4EN2	English	40	60	100	4	-	-	3	3
	21BSC2L4HI2	Hindi								
	21BSC2L4SN2	Sanskrit								
	21BSC2L4TE2	Telugu								
	21BSC2L4UR2	Urdu								
DSC2	21BSC2C2CS2L	Data Structures using C	40	60	100	4	-	-	4	3
	21BSC2C2CS2P	Data structures Lab	25	25	50	-	-	4	2	3
DSC2	Another Department Code	Another Department Course Title	40	60	100	4	-	-	4	3
			25	25	50	-	-	4	2	3
AECC1	21BSC2AE1ES2	Environmental Studies	20	30	50	1	-	2	2	2
VBC3	21BSC2V3PE2	Physical Education- Sports	25	-	25	-	-	2	1	-
VBC4	21BSC2V4NC1	NCC/NSS/R&R(S&G) / Cultural	25	-	25	-	-	2	1	-
OEC2	21BSC2O2CS2	Web Designing	40	60	100	3	-	-	3	3
		Total Marks			**700**	**Semester Credits**			**25**	

Category	Course code	Title of the Paper	Marks			Teaching hours/week			Credit	Duration of exams (Hrs)
			IA	SEE	Total	L	T	P		
L5	21BSC3L5LK3	Kannada	40	60	100	4	-	-	3	3
	21BSC3L5LFK3	Functional Kannada								
L6	21BSC3L6EN3	English	40	60	100	4	-	-	3	3
	21BSC3L6HI3	Hindi								
	21BSC3L6SN3	Sanskrit								
	21BSC3L6TE3	Telugu								
	21BSC3L6UR3	Urdu								
DSC3	21BSC3C3CS1L	Object Oriented Programming in JAVA	40	60	100	4	-	-	4	3
	21BSC3C3CS1P	JAVA Lab	25	25	50	-	-	4	2	3
DSC3	Another Department Code	Another Department Course Title	40	60	100	4	-	-	4	3
			25	25	50	-	-	4	2	3
SEC2	21BSC3SE2ES2	Artificial Intelligence	25	25	50	1	-	2	2	2
VBC5	21BSC3V5PE3	Physical Education- Sports	25	-	25	-	-	2	1	-
VBC6	21BSC3V6NC2	NCC/NSS/R&R(S&G) / Cultural	25	-	25	-	-	2	1	-
OEC3	21BSC3O3CS5	E-Commerce	40	60	100	3	-	-	3	3
		Total Marks	700			Semester Credits			25	

Category	Course code	Title of the Paper	Marks			Teaching hours/week			Credit	Duration of exams (Hrs)
			IA	SEE	Total	L	T	P		
L7	21BSC4L7LK4	Kannada	40	60	100	4	-	-	3	3
	21BSC4L7LFK4	Functional Kannada								
L8	21BSC4L8EN4	English	40	60	100	4	-	-	3	3
	21BSC4L8HI4	Hindi								
	21BSC4L8SN4	Sanskrit								
	21BSC4L8TE4	Telugu								
	21BSC4L8UR4	Urdu								
DSC4	21BSC4C2CS2L	Database Management Systems	40	60	100	4	-	-	4	3
	21BSC4C2CS2P	DBMS Lab	25	25	50	-	-	4	2	3
DSC4	Another Department Code	Another Department Course Title	40	60	100	4	-	-	4	3
			25	25	50	-	-	4	2	3
AECC2	21BSC4AE1ES2	Constitution of India	25	25	50	1	-	2	2	2
VBC7	21BSC4V5PE4	Physical Education- Sports	25	-	25	-	-	2	1	-
VBC8	21BSC4V6NC3	NCC/NSS/R&R(S&G) / Cultural	25	-	25	-	-	2	1	-
OEC4	21BSC4O3CS7	Office Automation	40	60	100	3	-	-	3	3
		Total Marks	700			Semester Credits			25	

		SEMESTER-V								
Category	**Course code**	**Title of the Paper**	**Marks**			**Teaching hours/week**			**Credit**	**Duration of exams (Hrs)**
			IA	**SEE**	**Total**	**L**	**T**	**P**		
Computer Science as Major Discipline										
DSC5	21BSC5C5CSMJ1L	Programming in PYTHON	40	60	100	3	-	-	3	3
	21BSC5C5CSMJ1P	PYTHON Programming lab	25	25	50	-	-	4	2	3
DSC6	21BSC5C5CSMJ2L	Computer Networks	40	60	100	3	-	-	3	3
	21BSC5C5CSMJ2P	Computer Networks Lab	25	25	50	-	-	4	2	3
DSC5	Another Department Code as a Minor Subject	Another Department Course Title	40	60	100	3	-	-	3	3
			25	25	50	-	-	4	2	3
VC1	21BSC5VC1US	Unix & Shell Programming	40	60	100	3	-	-	3	3
	21BSC5VC1FD	Fundamentals of Data Science								
VBC9	21BSC5V5PE5	Physical Education-Sports	25	-	25	-	-	2	1	-
VBC10	21BSC5V6NC4	NCC/NSS/R&R(S&G) / Cultural	25	-	25	-	-	2	1	-
SEC3	21BSC5SE3CS3	Cyber Security	25	25	50	1	-	2	2	2
		Total Marks	**700**			**Semester Credits**			**22**	

Category	Course code	Title of the Paper	Marks			Teaching hours/week			Credit	Duration of exams (Hrs)
			IA	SEE	Total	L	T	P		
SEMESTER-VI										
Computer Science as Major Discipline										
DSC7	21BSC6C6CSMJ1L	Internet Technologies	40	60	100	3	-	-	3	3
	21BSC6C6CSMJ1P	Internet Technology Lab	25	25	50	-	-	4	2	3
DSC8	21BSC6C6CSMJ2L	Operating System Concepts	40	60	100	3	-	-	3	3
	21BSC6C6CSMJ2P	C# Programming Lab	25	25	50	-	-	4	2	3
DSC6	Another Department Code as a Minor Subject	Another Department Course Title	40	60	100	3	-	-	3	3
			25	25	50	-	-	4	2	3
VC2	21BSC6VC2HT	Health Care Technologies	40	60	100	3	-	-	3	3
	21BSC6VC2DM	Digital Marketing								
INT1	21BSC6 INT1L	Internship	25	25	50	-	-	2	2	2
VBC1	21BSC6V5PE5	Physical Education- Sports	25	-	25	-	-	2	1	-
VBC2	21BSC6V6NC4	NCC/NSS/R&R(S&G) / Cultural	25	-	25	-	-	2	1	-
SEC4	21BSC6SE4CS4	Professional Communication	25	25	50	1	-	2	2	2
Total Marks			**700**			**Semester Credits**			**24**	
Total Marks for BSC Program			**4200**			**Total Credits for BSC Program**			**146**	

Computer Science Subject as a Minor Discipline

SEMESTER-V										
Category	**Course code**	**Title of the Paper**	**Marks**			**Teaching hours/week**			**Credit**	**Duration of exams (Hrs)**
			IA	SEE	Total	L	T	P		
DSC5 As a Minor Subject	21BSC5C5CSMN1L	Programming in PYTHON	40	60	100	3	-	-	3	3
	21BSC5C5CSMN1P	PYTHON Programming lab	25	25	50	-	-	4	2	3

SEMESTER-VI										
Category	**Course code**	**Title of the Paper**	**Marks**			**Teaching hours/week**			**Credit**	**Duration of exams (Hrs)**
			IA	SEE	Total	L	T	P		
DSC6 As a Minor Subject	21BSC6C6CSMN1L	Internet Technologies	40	60	100	3	-	-	3	3
	21BSC6C6CSMN1P	Internet Technologies Lab	25	25	50	-	-	4	2	3

Exit Option

Choice Based Credit System [CBCS] of Computer Science Major & One Minor Discipline Scheme for the Four Years Computer Science B.Sc. Undergraduate Honors Programme with effect from 2021-22

Sl. No	Years	After completion of	Exit Option
1.	First	I and II **Semesters**	UG Certificate Course in Computer Science
2.	Second	III and IV **Semesters**	UG Diploma in Computer Science
3.	Third	V and VI **Semesters**	B.Sc in Computer Science
4.	Fourth	VII and VIII **Semesters**	B.Sc. (Hons.) in Computer Science

Concept Note, Abbreviation Explanation and Coding:

Concept Note:

1. **CBCS** is a mode of learning in higher education which facilitates a student to have some freedom in selecting his/her own choices, across various disciplines for completing a UG/PG program.
2. A credit is a unit of study of a fixed duration. For the purpose of computation of workload as per UGC norms the following is mechanism be adopted in the University:
    One credit (01) = One Theory Lecture (L) period of one (1) hour.
    One credit (01) = One Tutorial (T) period of one (1) hour.
    One credit (01) = One practical (P) period of two (2) hours.
3. Course: paper/subject associated with AECC, DSC, DSEC, SEC, VBC, OEC, VC, IC and MIL
4. In case of **B.Sc**. **Once a candidate chose two courses/subjects of a particular two department in the beginning, he/she shall continue the same till the end of the degree, then there is no provision to change the course(s) and Department(s).**
5. A candidate shall choose **one of the Department's courses as major and other Department course as minor in fifth and sixth semester and major course will get continued in higher semester.**
6. Wherever there is a practical there will be no tutorial and vice-versa
7. A major subject is the subject that's the main focus of Core degree/concerned.
8. A minor is a secondary choice of subject that complements core major/ concerned.
9. Vocational course is a course that enables individual to acquire skills set that are required for a particular job.
10. Internship is a designated activity that carries some credits involving more than **25 days** of working in an organization (either in same organization or outside) under the guidance of an identified mentor. Internship shall be an integral part of the curriculum.
11. **OEC: For non- computer science students. Computer Science students have to opt for OEC from departments other than major and minor disciplines.**

Abbreviation Explanations:

1. AECC: Ability Enhancement Compulsory Course.
2. DSC: Discipline Specific Core Course.
3. DSEC: Discipline Specific Elective Course.
4. SEC: Skill Enhancement Course.
5. VBC: Value Based Course.
6. OEC: Open/Generic Elective Course.
7. VC: Vocational Course.
8. IC: Internship Course.
9. L1:   Language One.
10. L2:   MIL .
11. L= Lecture; T= Tutorial; P=Practical.
12. MIL= Modern Indian Language; English or Hindi or Telugu or Sanskrit or Urdu.

Program Coding:
1. Code 21: Year of Implementation.
2. Code BSC: BSC Program under the faculty of Applied Science of the University.
3. Code 1: First Semester of the Program, (2 to 6 represent higher semesters).
4. Code AE: AECC, (C for DSC, S for SEC, V for VBC and O for OEC).
5. Code 1: First "AECC" Course in semester, similarly in remaining semester for such other courses.
6. Code LK: Language Kannada, similarly Language English, Language Hindi, Language Telugu, Language Sanskrit, &Language Urdu.
7. Code 1: Course in that semester.
8. CS: Computer Science.

Evaluation Scheme for Lab Examination

Assessment Criteria		25 marks
Program – 1 from Part A	Writing the Program	05
	Execution and Formatting	05
Program -2 from Part B	Writing the Program	05
	Execution and Formatting	05
Viva Voice		05
Total		**25**

ASSESSMENT METHODS
Evaluation Scheme for Internal Assessment:

Theory:

Assessment Criteria	40 marks
1st Internal Assessment Test for30 marks 1 hr 30 min after 8 weeks and 2nd Internal Assessment Test for 30 marks 1 hr 30 min after 15 weeks. Average of two tests should be considered.	30
Attendance >75%	05
Assignment	05
Total	**40**

Practical:

Assessment Criteria	25 marks
Semester End Internal Assessment Test for 15 marks 2 hrs	15
Attendance >75%	05
Journal (Practical Record)	05
Total	**25**

BSc Question Paper Pattern

Time: 3 Hrs   **PART-A**   **Max Marks. 60**

Answer any Five questions.   **5X2=10**

    1.

    2.

    3.

    4

    5

    6

    7

    8

Note: Two questions from each unit.

PART-B

Answer any <u>Five</u> of the following questions.   **5X4=20**

    1

    2

    3

    4

    5

    6

    7

    8

Note: Two questions from each unit.

PART-C

Answer any <u>Three</u> of the following questions.   **3X10=30**

    1

    2

    3

    4

    5

Note: Minimum One question from each unit.

Semester I

Year	I	**Course Code:** 21BSC1C1CS1L	**Credits**	04
Sem.	I	**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:.03 hrs.	

Course Outcomes	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 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 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 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 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	13

	operator; Special operators; Operator Precedence and Associatively; Evaluation of arithmetic expressions; Type conversion.	
Unit III	**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. **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.	13
Unit IV	**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
Recommended Leaning Resources	
Print Resources	**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 & Ritche: 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**	52

Course Pre-requisites, if any:	Knowledge of Programming	
Formative Assessment Marks: 25	Summative Assessment Marks: 25	Duration of ESA: 03 hrs.

Practice Labs

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 *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 for other discipline

Open Elective 1: C Programming Concepts

Year	I	**Course Code:** 21BSC1O1CS1		**Credits**	03
Sem.	1	**Course Title: C Programming Concepts**		**Hours**	42
Course Pre-requisites, if any		NA			
Formative Assessment Marks: 40		Summative Assessment Marks: 60		Duration of ESA:.03 hrs.	

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 -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.	6Hrs
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. **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.	10 Hrs
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 Associatively; Evaluation of arithmetic expressions; Type conversion. **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.	8 Hrs
Unit - 4	
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.	8 Hrs
Unit 5:	

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;	10Hrs

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)

Year	I	**Course Code:** 21BSC2C2CS2L		**Credits**	04
Sem.	2	**Course Title:** Data Structures using C		**Hours**	52
Course Pre-requisites, if any		NA			
Formative Assessment Marks: 40		Summative Assessment Marks: 60		Duration of ESA: 03 hrs.	

Course Outcomes	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

Unit No.	Course Content	Hours
Unit I	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. Arrays: Basic Concepts – Definition, Declaration, Initialisation, Operations on arrays; Types of arrays; Arrays as abstract data types (ADT); Representation of Linear Arrays in memory;	13
Unit II	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; 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;	13
Unit III	Queues: Basic Concepts – Definition and Representation of queues;	13

	Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues; Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de- allocation functions - malloc, calloc, reallocand free. 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	
Unit IV	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
Recommended Leaning Resources		
Print Resources	**Reference Books:** 1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures 2. Tanenbaum: Data structures using C (Pearson Education) 3. Kamathane: Introduction to Data structures (Pearson Education) 4. Y. Kanitkar: Data Structures Using C(BPB) 5. Kottur: Data Structure Using C 6. Padma Reddy: Data Structure Using C 7. Sudipa Mukherjee: Data Structures using C – 1000 Problems and Solutions (McGraw Hill Education,2007)	

Year	I	**Course Code:** 21BSC2C2CS2P		**Credits**	02
Sem.	I	**Course Title:** Data Structure Lab		**Hours**	52

Course Pre-requisites, if any:	Knowledge of Programming	
Formative Assessment Marks: 25	Summative Assessment Marks: 25	Duration of ESA: 03 hrs.

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.

Open Elective Courses offered by the Department of Computer Science for other discipline

Open Elective 2: Web Designing

Year	I	**Course Code:** 21BSC2O2CS2		**Credits**	03
Sem.	II	**Course Title: Web Designing**		**Hours**	42
Course Pre-requisites, if any		NA			
Formative Assessment Marks: 40		Summative Assessment Marks: 60		Duration of ESA:.03 hrs.	

Course Outcomes (COs):

- Be familiar with different web design theories and terminology.
- Analyze a web page and identify its elements and attributes.
- Create web pages using XHTML and Cascading Style Sheets.
- Build dynamic web pages using JavaScript (Client-side programming).

Unit – 1	
Fundamentals: Internet, WWW, Web Browsers and Web Servers, URLs, MIME, HTTP, Security, the Web Programmers Toolbox. Introduction to XHTML: Basic syntax, Standard structure, Basic text markup, Images, Hypertext Links, Lists, Tables, Forms, Frames	8 Hrs
Unit-2	
Cascading Style Sheet (CSS): Introduction, Levels of style sheets, Style specification formats, Selector forms, Property value forms, Font properties, List properties, Color, Alignment of text, The box model, Background images, The <span> and <div> tags, Conflict resolution.	8 Hrs
Unit-3	
The Basics of JavaScript: Overview of JavaScript, Object orientation and JavaScript, Syntactic characteristics, Primitives, operations, and expressions, Screen output and keyboard input, Control statements, Object creation and modification, Arrays, Functions, Constructors, Pattern matching using regular expressions, Errors in scripts, Examples.	10Hrs
Unit-4	
JavaScript and HTML Documents: The JavaScript execution environment, The Document Object Model, Element access in JavaScript, Events and event handling, handling events from the Body elements, Button elements, Text box and Password elements, The DOM 2 event model, the navigator object, DOM tree traversal and modification.	8 Hrs

Unit-5	
Dynamic documents with JavaScript: Introduction, positioning elements, Moving elements, Element visibility, Changing colors and fonts, Dynamic content, Stacking elements, locating the mouse cursor, Reacting to a mouse click, slow movement of elements, Dragging and dropping elements.	8 Hrs

Text Books:

1. Robert W. Sebesta: Programming the World Wide Web,4th Edition, Pearson Education, 2008.

References:

1. M. Deitel, P.J. Deitel, A. B. Goldberg: Internet & World Wide Web How to Program, 4th Edition, Pearson Education, 2004.
2. Chris Bates: Web Programming Building Internet Applications,3rd Edition, Wiley India, 2007.
3. Xue Bai et al: The web Warrior Guide to Web Programming, Cengage Learning, 2003.
4. M Srinivasan: Web