Telephone No. 2419677/2419361
Fax: 0821-2419363/2419301

e-mail : registrar@uni-mysore.ac.in
www.uni-mysore.ac.in

UNIVERSITY OF MYSORE

Estd. 1916

VishwavidyanilayaKaryasoudha
Crawford Hall, Mysuru- 570 005

No.AC2(S)/151/2020-21

Dated: 26-10-2021

Notification

Sub:- Syllabus and Examination Pattern of Computer Science (UG) with effective from the Academic year 2021-22 as per NEP-2020.

Ref:-
1. Decision of Board of Studies in Computer Science (UG) meeting held on 29-09-2021 & 30-09-2021.
2. Decision of the Faculty of Science & Technology Meeting held on 16-10-2021.
3. Decision of the Academic Council meeting held on 22-10-2021.

*****

The Board of studies in Computer Science (UG) which met on 29-09-2021 & 30-09-2021 has recommended & approved the syllabus and pattern of Examination of Computer Science Programme with effective from the Academic year 2021-22 as per NEP -2020.

The Faculty of Science & Technology and Academic Council at their meetings held on 16-10-2021 and 22-10-2021 respectively have also approved the above said proposal and it is hereby notified.

The syllabus and Examination pattern is annexed herewith and the contents may be downloaded from the University Website i.e., www.uni-mysore.ac.in.

Registrar

Registrar
University of Mysore
Mysore

To:-

1. All the Principal of affiliated Colleges of University of Mysore, Mysore. Those who are running B.Sc Courses.
2. The Registrar (Evaluation), University of Mysore, Mysuru.
3. The Chairman, BOS/DOS, in Computer Sciences, Manasagangothri, Mysore.
4. The Dean, Faculty of Science & Technology, DoS in Psychology, MGM.
5. The Director, Distance Education Programme, Moulya Bhavan, Manasagangotri, Mysuru.
6. The Director, PMEB, Manasagangothri, Mysore.
7. Director, College Development Council , Manasagangothri, Mysore.
8. The Deputy Registrar/Assistant Registrar/Superintendent, Administrative Branch and Examination Branch, University of Mysore, Mysuru.
9. The PA to Vice-Chancellor/ Registrar/ Registrar (Evaluation), University of Mysore, Mysuru.

Annexure-B.Sc (CS) Scheme

NEP 2020 CURRICULUM

B.Sc(Computer Science)

The objectives of the 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 development of 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 ofa given information handling problem.
5. The ability to work independently on a substantial software project and as an effective team member.

Curriculum Structure

Program: B.Sc. (Basic and Honors) Subject: Computer Science

1. Computer Science as MAJOR with another Subject as MINOR (Table IIA of Model Curriculum)

Sem	Discipline Specific Core Courses (DSC)	Hour of Teaching/ Week		Discipline Specific Elective Courses (DSE)/ Vocational Courses (VC)	Hour of Teaching/ Week
		Theory	Lab		
1	DSC-1: Computer Fundamentals and Programming in C DSC-1Lab: C Programming Lab	4	 4		
2	DSC-2: Data Structures using C DSC-2Lab: Data structures Lab	4	 4		
3	DSC-3: Object Oriented Programming Concepts and Programming in JAVA DSC-3Lab: JAVA Lab	4	 4		
4	DSC-4: Database Management Systems DSC-4Lab: DBMS Lab	4	 4		
5	DSC-5: Programming in PYTHON DSC-6: Computer Networks DSC-5Lab: PYTHON Programming lab DSC-6Lab: Computer Networks Lab	3 3	 4 4	VC-1: Any one from Vocational Courses, Group – 1*	3
6	DSC-7: Internet Technologies DSC-8: Operating System Concepts DSC-7Lab: JAVA Script, HTML, CSS Lab DSC-8Lab: C# Programming Lab	3 3	 4 4	VC-2: Any one from Vocational Courses, Group – 2* Internship:	3 2
7	DSC-9: Computer Graphics and Visualization DSC-10: Design and Analysis of Algorithms DSC-11: Software Engineering DSC-9Lab: Computer Graphics and Visualization Lab DSC-10Lab: Algorithms Lab	3 3 3	 4 4	DSE-1: Any one from Discipline Specific Elective Courses, Group – 1** DSE-2: Any one from Discipline Specific Elective Courses, Group – 2** Research Methodology:	3 3 3
8	DSC-12: Artificial Intelligence and Applications DSC-13: Computer Organization and Architecture DSC-14: Data Warehousing and Data Mining DSC-12Lab: AI Lab	3 3 3	 4	DSE-3: Any one from Discipline Specific Elective Courses, Group – 3** Research Project:	3 6

2. Computer Science as MAJOR with another Subject also as MAJOR
(Table IIIA of Model Curriculum)

Sem	Discipline Specific Core Courses (DSC)	Hour of Teaching/ Week		Discipline Specific Elective Courses (DSE)	Hour of Teaching/ Week
		Theory	Lab		
1	DSC-1: Computer Fundamentals and Programming in C DSC-1Lab: C Programming Lab	4	 4		
2	DSC-2: Data Structures using C DSC-2Lab: Data structures Lab	4	 4		
3	DSC-3: Object Oriented Programming Concepts and Programming in JAVA DSC-3Lab: JAVA Lab	4	 4		
4	DSC-4: Database Management Systems DSC-4Lab: DBMS Lab	4	 4		
5	DSC-5: Programming in PYTHON DSC-6: Computer Networks DSC-5Lab: PYTHON Programming lab DSC-6Lab: Computer Networks Lab	3 3	 4 4		
6	DSC-7: Internet Technologies DSC-8: Operating System Concepts DSC-7Lab: JAVA Script, HTML, CSS Lab DSC-8Lab: C# Programming Lab	3 3	 4 4		
7	DSC-9: Computer Graphics and Visualization DSC-10: Design and Analysis of Algorithms DSC-11: Software Engineering DSC-9Lab: Computer Graphics and Visualization Lab DSC-10Lab: Algorithms Lab	3 3 3	 4 4	DSE-1: Any one from Discipline Specific Elective Courses, Group – 1** DSE-2: Any one from Discipline Specific Elective Courses, Group – 2** Research Methodology:	3 3 3
8	DSC-12: Artificial Intelligence and Applications DSC-13: Computer Organization and Architecture DSC-14: Data Warehousing and Data Mining DSC-12Lab: AI Lab	3 3 3	 4	DSE-3: Any one from Discipline Specific Elective Courses, Group – 3** Research Project:	3 6

3. Computer Science as MINOR with another Subject as MAJOR (As per Table IIA of Model Curriculum)

Sem	Discipline Specific Core Courses (DSC)	Hour of Teaching/ Week	
		Theory	Lab
1	DSC-1: Computer Fundamentals and Programming in C DSC-1Lab: C Programming Lab	4	4
2	DSC-2: Data Structures using C DSC-2Lab: Data structures Lab	4	4
3	DSC-3: Object Oriented Programming Concepts and Programming in JAVA DSC-3Lab: JAVA Lab	4	4
4	DSC-4: Database Management Systems DSC-4Lab: DBMS Lab	4	4
5	DSC-5: Programming in PYTHON DSC-5Lab: PYTHON Programming lab	3	4
6	DSC-6: Internet Technologies DSC-6Lab: JAVA Script, HTML, CSS Lab	3	4

*** Vocational Courses:**

Group-1:	Group-2:
• DTP, CAD and Multimedia • Hardware and Server Maintenance • Web Content Management Systems • E-Commerce • Web Designing	• Health Care Technologies • Digital Marketing • Office Automation • Multimedia Processing • Accounting Package

**** Discipline Specific Elective Courses:**

Group-1:	Group-2:	Group-3:
• IoT • Cyber Law and Cyber Security • Web Programming - PHP and MySQL • Clouds, Grids, and Clusters • Software Testing	• Information and Network Security • Data Compression • Discrete Structures • Opensource Programming • Multimedia Computing • Big Data	• Data Analytics • Storage Area Networks • Pattern Recognition • Digital Image Processing • Parallel Programming • Digital Signal Processing

Annexure: B.Sc Syllabus

Syllabus for B.Sc (Basic and Honors) 1st and 2nd Semesters

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: 02

Course Outcomes (COs):

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

- Confidently operate 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, strings , structures, unions and functions

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. **C Programming Basic Concepts:** C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants.	13
Unit - 2	
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. 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 - 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. M.T Somashekara, D.S Guru and K.S. Manjunatha: Problem solving with C, PHI publication
3. 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-1Lab	**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 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	04
	Execution and Formatting	04
Program -2 from Part B	Flowchart/Algorithm	02
	Writing the Program	04
	Execution and Formatting	04
Viva Voce based on C Programming		02
Practical Record		03
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: 02 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 ${}^{n}C_{r}$, 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 - 2	
Operations on arrays : 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. 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. Queues: Basic Concepts – Definition and Representation of queues; Types of queues – Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues;	13

Unit - 3	
Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - malloc, calloc, realloc and 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	13
Unit - 4	
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. Sartaj 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. 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	04
	Execution and Formatting	04
Program -2 from Part B	Flowchart/Algorithm	02
	Writing the Program	04
	Execution and Formatting	04
Viva Voce based on C Programming		02
Practical Record		03
Total		**25**

Note: The syllabi of the courses of remaining semesters shall be framed in subsequent BoS meetings.