Semester II

C Programming Lab

(Stream A & B)

Course Code	Teaching hours per week				Credits	Exam Hours	Course Type	CIE Marks	ESE Marks
	L	T	P	R					
26ABNEL201	0	0	2	0	2	2 Hrs. 30 Mins	Lab	50	50

1. **Preamble: The C Programming Laboratory is designed to develop problem-solving and computational thinking skills through the C programming language. The course focuses on algorithm design, logical reasoning, structured programming and modular software development. Students will gain hands-on experience in designing, implementing, testing and debugging C programs to develop efficient solutions for computational and real-world problems.**

2. **Prerequisites : Nil**

List of Experiments

Experiment Number	Syllabus Description: List of Experiments
1	**Introduction to C Programming and Computational Problem Solving.** **Objectives:** ● **Familiarize students with the C programming environment.** ● **Understand the structure of a C program.** ● **Introduce compilation, execution and debugging.** ● **Implement basic computational problems using variables, data types, expressions, operators and input/output operations.** **Sample List of Experiments :** 1. **Display formatted output.** 2. **Temperature conversion.** 3. **Distance between two points.** 4. **Unit conversion problems.** 5. **Simple arithmetic computation applications.** 6. **Character conversion problems.**

2	**Decision Making and Conditional Logic.** **Objectives:** ● **Develop programs involving decision-making and conditional execution.** ● **Implement computational problems using relational, logical, and conditional operators.** ● **Apply selection control structures (if-else, nested conditions and conditional operator) for solving numerical and real-world problems.** **Sample List of Experiments :** 1. **Odd or even number classification** 2. **Positive/negative/zero classification** 3. **Largest among numbers** 4. **Divisibility checking** 5. **Leap year verification** 6. **Quadratic equation roots** 7. **Character type identification (alphabet/digit/special character)** 8. **Vowel or consonant identification** 9. **Uppercase/lowercase character classification** 10. **Grade assignment based on score ranges**
3	**Iterative Problem Solving.** **Objectives:** ● **Develop programs involving repetitive computations using iterative constructs.** ● **Apply looping structures (for loop, while loop and do-while loop) for solving mathematical and pattern-based problems.** ● **Implement number-processing applications using nested loops and control statements.** **Sample List of Experiments :** 1. **Factorial of a number** 2. **Prime number verification** 3. **Palindrome checking** 4. **Perfect number detection** 5. **Pattern/pyramid generation**

4	**Array-Based Data Processing.** **Objectives:** ● **Develop programs for storing and processing collections of data using one-dimensional arrays.** ● **Implement searching, sorting and array manipulation techniques.** ● **Apply array traversal and function-based operations for solving data-processing problems.** **Sample List of Experiments :** 1. **Sequential/Linear search** 2. **Bubble sort** 3. **Array rotation** 4. **Frequency/count-based problems (maximum frequency, duplicate elements, occurrence of character etc).**
5	**Matrix Operations and Multidimensional Arrays.** **Objectives:** ● **Develop programs using multidimensional arrays for matrix operations.** ● **Implement matrix manipulation techniques using nested loops and array indexing.** ● **Apply multidimensional array operations for computational problem solving.** **Sample List of Experiments :** 1. **Matrix addition** 2. **Matrix multiplication** 3. **Matrix transpose**
6	**String Processing Applications.** **Objectives:** ● **Develop programs for processing and manipulating strings using character arrays.** ● **Apply string handling operations and standard string functions for text-based computational problems.** ● **Implement text-processing utilities using character manipulation techniques.** **Sample List of Experiments :** 1. **String palindrome checking** 2. **String sorting** 3. **Character counting** 4. **Text manipulation problems (reverse, concatenate, copy, case conversion, replace etc)**

7	**Functional Decomposition and Modular Programming. Objectives:** ● Develop modular programs using user-defined functions. ● Apply function declaration, function definition, parameter passing, return statements and recursion for problem solving. ● Implement computational problems by decomposing them into smaller functional units. **Sample List of Experiments :** 1. Fibonacci series 2. Mathematical series (sine, cosine, exponential, sum of fibonacci etc) 3. Recursive computations (fibonacci, factorial, gcd etc using recursive function) 4. Menu-driven applications
8	**Structured Data Representation.** **Objectives:** ● Develop programs using structures for organizing heterogeneous data. ● Implement record management applications using arrays of structures. ● Apply structure manipulation and sorting techniques for structured data processing. **Sample List of Experiments :** 1. Employee payroll system: Implement a C program that reads employee's ID, name, status('F' for full-time and 'P' for part-time) and basic pay. DA and HRA for full-time employees are 75% and 10% of basic pay respectively. Part-time employees have basic pay only. Print the total salary of the employee. 2. Vehicle information management system: Define a structure named Vehicle with members, model (character array), year (integer) and price(float). Create an array of 10 Vehicle structures and write a function to accept user input for each vehicle's details and sort the array in descending order based on price.

9	**Pointer-Based Programming.** **Objectives:** ● **Develop programs using pointers for indirect data access and manipulation.** ● **Apply pointer operations, address manipulation and parameter passing techniques for problem solving.** ● **Implement simple applications using pointer-based operations.** **Sample List of Experiments :** 1. **Swapping using pointers** 2. **Pointer-based array manipulation**
10	**File Handling and Persistent Data Storage.** **Objectives:** ● **Develop programs for permanent storage and retrieval of data using files.** ● **Implement file operations including file creation, reading, writing and appending data.** ● **Apply file handling functions for simple data management applications.** **Sample List of Experiments :** 1. **File creation and data management operations** 2. **Writing/appending/displaying records**

11	**Micro Projects.** **Students shall undertake simple micro projects based on real-world and socially relevant applications using C programming concepts. The micro projects are intended to help students apply problem-solving skills, structured programming concepts and computational thinking to practical applications.** **Guidelines for Micro Project Execution:** ● **Micro projects shall be assigned during the course of the laboratory.** ● **Students are expected to:** ○ **Analyze the given problem and identify suitable programming constructs.** ○ **Develop algorithms and modular solutions for the problem.** ○ **Implement the solution using appropriate C programming concepts.** ○ **Test and debug the developed application.** ● **Emphasis shall be on logical problem solving, modular program development and application-oriented implementation.** **Sample Micro Project Topics:** **1. Electricity Bill Generation System** **Develop a program to compute electricity bills based on consumer details and unit consumption. The application shall generate bills using different tariff ranges and displays the bill summary.** **2. Simple Banking Management System** **Develop a menu-driven banking application to perform operations such as deposit, withdrawal, balance enquiry, and transaction display for multiple customers.** **3. Library Book Management System** **Develop a program to maintain basic library records including book title, author, issue status, and availability. The system shall support searching and updating book details.** **4. Community Donation Record System** **Develop a simple application to maintain records of donated items or funds**

	collected for community service activities. The system shall support data entry, updating, searching and report generation. 5. Inventory Management System Develop a program to maintain product details, stock availability and billing information for a small store or canteen.

3. Course Assessment Method
(CIE: 50 marks, ESE: 50 marks)

Continuous Internal Evaluation Marks (CIE) :

Attendance	Preparation/Pre-Lab Work experiments, Viva and Timely completion of Lab Reports / Record (Continuous Assessment)	Micro Projects	Internal Examination	Total
5	20	10	15	50

End Semester Examination Marks (ESE):

Procedure/ Preparatory work/Design/ Algorithm	Conduct of experiment/ Execution of work/ troubleshooting/ Programming	Result with valid inference/ Quality of Output	Viva voce	Record	Total
10	15	10	10	5	50

Submission of Record: Students shall be allowed for the end semester examination only upon submitting the duly certified record.

4. Course Outcomes

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

Course Outcomes		Blooms Taxonomy Level
CO1	Apply basic C programming constructs, including control structures, to solve computational problems.	K3

CO2	**Construct C programs using arrays, matrices and strings.**	**K3**
CO3	**Apply functions and user-defined data types to develop structured C programs.**	**K3**
CO4	**Utilize pointers and file handling concepts in C programs for data management applications.**	**K3**
CO5	**Apply C programming constructs to develop solutions for real-world problems.**	**K3**

Note: K1- Remember, K2- Understand, K3- Apply, K4- Analyze, K5- Evaluate, K6- Create

5. **Mapping of Course Outcomes with Program Outcomes**

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11
CO1	3	2	-	-	2	-	-	-	-	-	-
CO2	3	2	2	-	2	-	-	-	-	-	-
CO3	3	2	2	-	2	-	-	-	-	-	-
CO4	3	2	2	-	2	-	-	-	-	-	-
CO5	3	3	3	2	3	-	-	2	2	2	2

Note: 1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High), - : No Correlation

6. **Text Books**

Sl No	Title of Book	Name of Author/s	Publisher	Edition and Year
1	Schaum's Outline of Programming with C	Byron S Gottfried	Mc Graw Hill Education (India)	2/e, 2022
2	Problem Solving and Program Design in C	Jeri R. Hanly, Elliot B. Koffman	Pearson	8/e, 2016
3	The C Programming Language	Brian W. Kernighan and Dennis Ritchie	Pearson	2/e, 2015

7. **Reference Books**

Sl No	Title of Book	Name of Author/s	Publisher	Edition and Year
1	Programming in ANSI C	E Balagurusamy	Mc Graw Hill	9/e, 2024

2	Programming in C	Reema Thareja	Oxford University Press	2/e, 2016
3	C The Complete Reference	Herbert Schildt	Mc Graw Hill	4/e, 2017
4	Let us C	Yashavant Kanetkar	BPB Publishers	19/e, 2022

8. Video Links or any other Reference Materials

Sl No	Details Like Weblink, Publication Details
1	https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs53
2	https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_cs40

9. Evaluation Pattern for Continuous Assessment (20 Marks):

- Preparation and Pre-Lab Work (5 Marks)
  - Pre-Lab Assignments : Assessment of pre-lab assignments or quizzes that test understanding of the upcoming experiment.
  - Understanding of Theory : Evaluation based on students' preparation and understanding of the theoretical background related to the experiments.
- Conduct of Experiments (6 Marks)
  - Procedure and Execution : Adherence to correct procedures, accurate execution of experiments and following safety protocols.
  - Skill Proficiency : Proficiency in handling equipment, accuracy in observations and troubleshooting skills during the experiments.
  - Teamwork: Collaboration and participation in group experiments.
- Lab Reports and Record Keeping (4 Marks)
  - Quality of Reports : Clarity, completeness and accuracy of lab reports. Proper documentation of experiments, data analysis and conclusions.
  - Timely Submission : Adhering to deadlines for submitting lab reports/rough records and maintaining a well-organized fair record.
- Viva Voce (5 Marks)
  - Oral Examination: Ability to explain the experiment, results and underlying principles during a viva voce session.

Final Marks Averaging: The final marks for preparation, conduct of experiments, viva and record are the average of all the specified experiments in the syllabus.

10. Evaluation Pattern for MicroProject (10 Marks)

- Design and Planning (2 Marks) :

Assessment of the understanding of the problem requirements, identification of objectives and formulation of an appropriate solution approach, design, algorithm, architecture or methodology as applicable.

- **Implementation and Functionality (5 Marks) :**

Assessment of the correctness, completeness, functionality and effectiveness of the implemented solution. Evaluation includes the appropriate application of concepts, techniques, tools and methodologies learned in the course to meet the specified requirements.

- Testing and Validation (1 Mark) :

Assessment of the verification and validation of the solution through appropriate test cases, experiments, observations, simulations or performance measures, as applicable and comparison of results with the expected outcomes.

- Documentation and Presentation (1 Mark) :

Assessment of the quality, organization and completeness of the project documentation, presentation of results and clarity of demonstration, including the ability to effectively communicate the problem, approach and outcomes.

- Viva Voce (1 Mark) :

Assessment of the student's understanding of the problem, design decisions, implementation methodology, results obtained and the underlying concepts and techniques employed in the microproject.

Note : A microproject shall be carried out individually or in teams to apply the concepts learned in the course for solving a given problem. The evaluation focuses on the design, implementation, testing, documentation and presentation of the solution. Originality of work and adherence to academic integrity are expected throughout the microproject activity.

11. Evaluation Pattern for End Semester Examination (50 Marks)

- **Procedure/Preliminary Work/Design/Algorithm (10 Marks)**
  - **Procedure Understanding and Description : Clarity in explaining the procedure and understanding each step involved.**
  - **Preliminary Work and Planning : Thoroughness in planning and organizing materials/equipment.**
  - **Algorithm Development : Correctness and efficiency of the algorithm related to the experiment.**
  - **Creativity and logic in algorithm or experimental design.**

- **Conduct of Experiment/Execution of Work/Programming (15 Marks)**

- **Setup and Execution: Proper setup and accurate execution of the experiment or programming task.**

- **Result with Valid Inference/Quality of Output (10 Marks)**
  - **Accuracy of Results : Precision and correctness of the obtained results.**
  - **Analysis and Interpretation : Validity of inferences drawn from the experiment or quality of program output.**

- **Viva Voce (10 Marks)**
  - **Ability to explain the experiment, procedure results and answer related questions.**
  - **Proficiency in answering questions related to theoretical and practical aspects of the subject.**

- **Record (5 Marks)**
  - **Completeness, clarity and accuracy of the lab record submitted.**

FISAT
Focus on Excellence