**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS**

**LESSION PLAN**

|  |  |  |  |
| --- | --- | --- | --- |
| Name of the Course Teacher | J. SEETHARAMAN | Designation, Department, Staff Code | Asst. Prof. / CSA |
| Degree & Branch | I BCA | Year / Semester | I / I |
| Course | Problem Solving Techniques | Credits | 3 |

**COURSE OBJECTIVES**

To learn the fundamentals of PST and methodologies which are essential for building good programs

To help the students to learn programming concepts using C Language

Design the solution from specification of a problem and write pseudo code of the algorithm

using basic building blocks or structured programming constructs

To demonstrate a thorough understanding of modular programming by designing programs which require the use of programmer-defined functions

To impart the knowledge about pointers which is the backbone of effective memory

handling and demonstrate adeptness of file access in developing solutions to problems

**COURSE OUTCOMES**

Learn the fundamental programming concepts and methodologies, which are essential for building good programs.

Able to translate an algorithm into program

Understand the modular programming by designing programs, which require the use of

control structures, arrays, defined functions and learn about storage classes

Acquire the knowledge about pointers

Obtain the knowledge of writing and testing programs

**TEXT BOOKS**

1. Venkatesh, Nagaraju Y, “Practical C Programming for Problem Solving”, Khanna Book Publishing Company, 2024.
2. R.S.Salaria, Programming for Problem Solving (with Lab Manual), Khanna Book

Publishing Company, 1st Edition, 2024.

1. Harvey Deitel and Paul Deitel, “C How to Program”, Pearson India, 9th Edition, 2015.
2. R G Dromey, “How to Solve It by Computer”, Pearson India, 2007.
3. Brian W. Kernighan and Dennis Ritchie, “The C Programming Language”, Pearson, 2nd Edition, 2015.

**E-REFERENCES:**

1. [https://practice.geeksforgeeks.org/explore/?category[]=C-Programming-Language](https://practice.geeksforgeeks.org/explore/?category%5b%5d=C-Programming-Language)
2. <https://www.codechef.com/practice>
3. <https://leetcode.com/problemset/all/>
4. https://exercism.org/tracks/c

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit I : Problems and Problem Instances (Week 1–3) 12 Hours**  **Learning Objectives:**   * Analysis of different types of Computational Problems and Solution Approaches. * Analysis and development of Algorithms * Problem solving steps with relevant and appropriate data structures   Types of Computational Problems- Classification of Problem- Analysis of Problems- Solution  Approaches- Algorithm Development- Analysis of Algorithm - Role of Data Structures in ProblemSolving- Problem Solving Steps-Modular Programming. | | | |
| **Week** | **Topics** | **Teaching Methods** | **Activities/Assessment** |
| 1 | Analysis different types of real time problems, types | Lecture & Discussion | Deriving real time problems |
| 1 | Deriving computational problems from scenarios with solutions | Lecture & Hands-on | Sample Problems and Solutions |
| 2 | Different types of Solution steps and Introduction to Algorithm | Examples & Practice | Steps involved in Solutions to sample problems |
| 2 | Algorithm steps and Introduction to Data Structures | Discussion and Hands-on | Comparison of different flows, blocks and data types |
| 3 | Data Structures in detail and Intro to Programming | Examples & Practice | Algorithm-Data structure mapping into Programming |
| 3 | Sample Problem solving with relevant Data structure, Algorithm and Program | Hands-on & Discussion | Sample programs |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit II: C Fundamentals (Week 4–6) – 12 Hours**  **Learning Objectives:**   * Understanding Basic blocks and structure of C Language. * Understand and apply core data structures in C.. * Understand functional programming & build in functions of C.   Character set - Identifier and keywords - data types - constants- Variables - Declarations -  Statements - Operators – Expressions – Built in functions – Input and Output Operations. | | | |
| **Week** | **Topics** | **Teaching Methods** | **Activities/Assessment** |
| 4 | Character set and types - Identifier and keywords, Hello World Program | Lecture & Discussion, Practice | Different types of char set, listing keywords with purpose, first C program |
| 4 | data types - constants- Variables - Declarations | Lecture & Discussion, Practice | Data types with right scenario, declaring const and variables rightly |
| 5 | Statements - Operators – Expressions | Lecture & Discussion, Practice | Practicing appropriate operator and statements |
| 5 | Built in functions | Lecture & Discussion, Practice | Practicing right usage of built in functions |
| 6 | Input and Output Operations. | Lecture & Discussion, Practice | Complete and simple C programs |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit III: Control Structure Statements and Arrays (Week 7–9) – 12 Hours**  **Learning Objectives:**   * Understanding and using different control, looping statements. * Understanding Arrays   Flow of control - if, if else, While, do-while, for loop, Nested control structures - Switch, break and continue, go to statement – Arrays. | | | |
| **Week** | **Topics** | **Teaching Methods** | **Activities/Assessment** |
| 7 | Control structure: If, If elseif | Theory + Practice | programs |
| 7 | Nested If and Intro to Looping statement | Theory + Practice | Code examples |
| 8 | While, do, for loops | Theory + Practice | Code examples |
| 8 | Continue, break, goto statements | Explanation + Tasks | Code examples |
| 9 | Arrays | Theory + Practice | Code examples |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit IV: Functions and Structures (Week 10–12) – 12 Hours**  **Learning Objectives:**   * Understand the need and usage of user defined functions with different parts. * Understand the need and usage of Structures and Unions   Definition - Prototypes – Passing arguments – Recursion- Storage Classes – Structure-Union | | | |
| **Week** | **Topics** | **Teaching Methods** | **Activities/Assessment** |
| 10 | Functions need, declaration, calling | Theory + Practice | Code examples |
| 11 | Diffent types arguments and recursive functions | Theory + Practice | Code examples |
| 12 | Structures and unions | Theory + Practice | Code examples |

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit V: Pointers and Files (Week 13–15) – 12 Hours**  **Learning Objectives:**   * Understand variable declaration, memory allocations and releasing with pointers * Understand persistence storage/retrieval with Files   Declarations - Structures and Pointers - Passing pointers to Functions - Pointer and Arrays –Arrays of Pointers - Files: Creating, Processing, Opening and Closing a data file- Debugging. | | | |
| **Week** | **Topics** | **Teaching Methods** | **Activities/Assessment** |
| 13 | Memory management and Pointers Intro | Theory + Practice | Code examples |
| 14 | Different types of pointers with Functions and Structures | Theory + Practice | Code examples |
| 15 | Store/retrieve data using File | Theory + Practice | Code examples |

**Assessment Strategy:**

* **Internal Assessment**: Periodic class tests, coding assignments, quizzes
* **Practical Exam**: Midterm and Final lab exams
* **Project**: Mini like billing and mark statements with files

-------------------------------------------------------------------------------------------------------------------------------

**Problem Solving Techniques LAB with C Language.**

**Weekly C Practical Lesson Plan (4 Hours/Week)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | **Experiment Topic** | **Learning Objectives** | **Lab Activities** | **Assessment** |
| **1,2** | Github Account creation and usage | - Introduction to version control system, github account creation with sample repo and file | -creating github repo with sample repo, reviewers and files | Verification of github repo with sample repo, reviewers and files |
| **3** | Converting degrees Celsius to Fahrenheit and vice versa | Understand variables, constants, input & output statements | Prog Converting degrees Celsius to Fahrenheit and vice versa | Verifying different type of variable declaration with appropriate data handling |
| **4** | Program to generate a Fibonacci Series | Understand control and looping statements | Program to generate a Fibonacci Series | Verifying usage of different control and looping statements |
| **5** | Given a positive integer value n (>= 0) display number, square and cube of numbers from  1 to n in a tabular format? | Understand control and looping statements | Given a positive integer value n (>= 0) display number, square and cube of numbers from  1 to n in a tabular format? | Verifying usage of different control and looping statements |
| **6** | Given an input positive integer number, display odd numbers from in the range [1,n] | Understand control and looping statements | Given an input positive integer number, display odd numbers from in the range [1,n] | Verifying usage of different control and looping statements |
| **7** | Compute character grade from the marks (0 ≤ marks ≤ 100) of a subject. Grading Scheme:  80-100: A, 60 - 79: B, 50 - 59: C, 40-49: D, 0-39: F? Solve this using switch case | Understand control and looping statements | Compute character grade from the marks (0 ≤ marks ≤ 100) of a subject. Grading Scheme:  80-100: A, 60 - 79: B, 50 - 59: C, 40-49: D, 0-39: F? Solve this using switch case | Verifying usage of different control and looping statements |
| **8** | Display following patterns of n rows (n > 0), For the below examples n = 5?  $  $$  $$$  $$$$  $$$$$ | Understand control and looping statements | Display following patterns of n rows (n > 0), For the below examples n = 5?  $  $$  $$$  $$$$  $$$$$ | Verifying usage of different control and looping statements |
| **9** | Check if a given positive integer number is a palindrome or not? | Understand control and looping statements | Check if a given positive integer number is a palindrome or not? | Understand control and looping statements |
| **10** | Implement your own string length and string reversal functions? | Understand control and looping statements | Implement your own string length and string reversal functions? | Understand control and looping statements |
| **11** | Write a Program to sort an array. | Understand control and looping statements with internal data structure functionality | Write a Program to sort an array. | Understand control and looping statements with internal data structure functionality |
| **12** | Write a Program to find the factorial of an integer using recursion. | Understand functions and recursive functions | Write a Program to find the factorial of an integer using recursion. | Understand functions and recursive functions |
| **13** | Write a Program to swap two values using function pointer | Understand memory management and pointers | Write a Program to swap two values using function pointer | Understand memory management and pointers |
| **14** | Write a Program to store student’s information using Structure | Understand pointers with data structures | Write a Program to store student’s information using Structure | Understand pointers with data structures |
| **15** | Write a Program to perform read and write operation on a file | Understand file storage and retrieval | Write a Program to perform read and write operation on a file | Understand file storage and retrieval |

**Notes:**

* **Each session** (4 hours) can be structured as:
  + **1 hour** – Concept briefing & demo
  + **2 hours** – Hands-on implementation
  + **1 hour** – Review, viva, and discussion
* **Assessment**: Conduct mini-viva, peer review, or evaluation rubrics weekly.
* Encourage students to **document their code** and maintain a **practical record**.

**Signature of the Course Teacher**

**Signature of the Head of the Department**