23CS1101 – INTRODUCTION TO PROGRAMMING

(Common to all Branches)

Course Category:	Professional core	**Credits:**	3
Course Type:	Theory	**Lecture - Tutorial - Practical:**	3-0-0
Prerequisite:	Knowledge on computer fundamentals and basic mathematics.	**Sessional Evaluation:** **Univ. Exam Evaluation:** **Total Marks:**	30 70 100
Objectives	To introduce students to the fundamentals of computer programming. To provide hands-on experience with coding and debugging. To foster logical thinking and problem-solving skills using programming. To familiarize students with programming concepts such as data types, control structures, functions, and arrays. To encourage collaborative learning and teamwork in coding projects		

Course Outcomes	A student after completion of the course will be able to	
	CO1	Describe the basics of computers, the concept of algorithm and algorithmic thinking.
	CO2	Analyse a problem and develop an algorithm to solve it.
	CO3	Write programs for various algorithms using the C programming language.
	CO4	Summarize more advanced features of C language
	CO5	Acquire problem-solving skills and the ability to debug and optimize the code.

Course Content

UNIT - I

Introduction to Programming and Problem Solving

History of Computers, Basic organization of a computer: ALU, input-output units, memory, program counter, Introduction to Programming Languages, Basics of a Computer Program, Algorithms, flowcharts (Using Dia Tool), pseudo code. Introduction to Compilation and Execution, Primitive Data Types, Variables, and Constants, Basic Input and Output Operations, Type Conversion, and Casting.

Problem solving techniques: Algorithmic approach, characteristics of algorithm, Problem solving strategies: Top-down approach, Bottom-up approach, Time and space complexities of algorithms

UNIT - II

Control Structures Simple sequential programs Conditional Statements (if, if-else, switch), Loops (for, while, dowhile) Break and Continue.

	UNIT - III **Arrays and Strings** Arrays indexing, memory model, programs with array of integers, two dimensional arrays, Introduction to Strings - Declaring Strings, Initializing Strings, Reading and Writing Strings, String Input / Output Functions, String Manipulation Functions. **UNIT - IV** **Pointers & User Defined Data types** Pointers, dereferencing and address operators, pointer and address arithmetic, array Manipulation using pointers, User-defined data types-Structures and Unions. **UNIT - V** **Functions & File Handling** Introduction to Functions, Function Declaration and Definition, Function call Return Types and Arguments, modifying parameters inside functions using pointers, arrays as parameters. Scope and Lifetime of Variables, Basics of File Handling
Text Books and References	**Text Books:** 1. Computer Science: A structured Programming Approach using C, B A Forouzan, Richard F Gilberg CENGAGE Learning, 3rd edition. 2. "The C Programming Language", Brian W. Kernighan and Dennis M. Ritchie, PrenticeHall, 1988 3. Schaum's Outline of Programming with C, Byron S Gottfried, McGraw-Hill Education, 1996
	Reference Books: 1. Computing fundamentals and C Programming, Balagurusamy, E., McGraw-Hill Education, 2008. 2. Programming in C, Rema Theraja, Oxford, 2016, 2nd edition
	Web Resource: 1. https://www.w3schools.com/ 2. https://www.onlinegdb.com/

23CS11P1 – COMPUTER PROGRAMMING LABORATORY

(Common to all Branches)

Course Category:	Professional Core	**Credits:**	1.5
Course Type:	Practical	**Lecture - Tutorial - Practical:**	0-0-3
Prerequisite:	Basic mathematical knowledge to solve problems and computer fundamentals.	**Sessional Evaluation:** **Univ. Exam Evaluation:** **Total Marks:**	30 70 100
Objectives	The course aims to give students hands – on experience and train them on the concepts of the C- programming language.		

Course Outcomes	Upon successful completion of the course, the students will be able:	
	CO1	Read, understand, and trace the execution of programs written in C language.
	CO2	Identify the right control structure for solving the problem.
	CO3	Implement C programs which utilize memory efficiently using programming constructs like pointers.
	CO4	Develop, Debug and Execute programs to demonstrate the applications of arrays, functions, basic concepts of pointers in C

Course Content

LIST OF EXPERIMENTS

Lab1: Familiarization with programming environment

i) Basic Linux environment and its editors like Vi, Vim & Emacs etc.

ii) Exposure to Turbo C, gcc

iii) Writing simple programs using printf(), scanf()

Lab 2: Converting algorithms/flow charts into C Source code. Developing the algorithms/flowcharts for the following sample programs

i) Sum and average of 3 numbers

ii) Conversion of Fahrenheit to Celsius and vice versa

iii) Simple interest calculation

Lab 3: Implement Programs on Data types, Operators, and Expressions.

Lab4: Demonstrate the significance of operator precedence and associativity in Expression Evaluation using C.

Lab 5: Implement decision-making constructs: if-else, goto, switch-case, break-continue

	in C. **Lab 6:** Demonstrate the scope of iterative constructs namely while loop, do-while loop and for loop in addition to structured jump constructs like break and continue using C programming. **Lab 7: (a)** Develop programs on one dimensional arrays and multidimensional arrays. **(b)** Implement Linear Search technique in C. **Lab 8: (a)** Implement various Matrix operations. **(b)** Implement String manipulation operations without predefined functions. **(c)** Implement Bubble Sort Technique **Lab 9:** Demonstrate Pointers & structures and memory dereference using C. **Lab10: (a)** Demonstrate the differences between structures and unions using a C program. **(b)** Create and display a singly linked list using self-referential structure. **(c)** Implement shift/ rotate operations on bit fields **Lab 11: (a)** Implement parameter passing technique: call-by-value. **(b)** Demonstrate scope and lifetime of variables in the context of subroutines and functions. **Lab 12:** Implement Recursive functions. **Lab 13:** (a) Implement parameter passing technique: call-by-reference. (b) Demonstrate the purpose of Dangling pointers using a C program. **Lab 14:** Implement various File operations (Open, Close, Read, Write, Append, and so on).
Text Books and References	**Text Books:** 1. Ajay Mittal, Programming in C: A practical approach, Pearson. 2. Byron Gottfried, Schaum' s Outline of Programming with C, McGraw Hill **Reference Books:** 1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India 2. C Programming, A Problem-Solving Approach, Forouzan, Gilberg, Prasad, CENGAGE
	Web Resources: 1. https://www.w3resource.com/c-programming-exercises/ 2. https://www.onlinegdb.com/

23IT12P2 – IT WORKSHOP

(Common to all branches of Engineering)

Course Category:	Professional Core	**Credits:**	1
Course Type:	Practical	**Lecture - Tutorial - Practical:**	0-0-2
Prerequisite:	Basics of Computer	**Sessional Evaluation:** **Univ. Exam Evaluation:** **Total Marks:**	30 70 100
Objectives	• To introduce the internal parts of a computer, peripherals, I/O ports, connecting cables • To demonstrate configuring the system as Dual boot both Windows and other Operating Systems Viz. Linux, BOSS • To teach basic command line interface commands on Linux. • To teach the usage of Internet for productivity and self-paced life-long learning • To introduce Compression, Multimedia and Antivirus tools and Office Tools such as Word processors, spread sheets and Presentation tools.		

Course Outcomes	At the end of the course, Student will be able to	
	CO1	Perform Hardware troubleshooting
	CO2	Describe Hardware components and inter dependencies.
	CO3	Safeguard computer systems from viruses/worms.
	CO4	Prepare Document/ Presentation.
	CO5	Prepare Document/ Presentation.

Course Content

LIST OF EXPERIMENTS

PC Hardware & Software Installation

Task 1: Identification of the peripherals of a computer, components in a CPU and its functions.

Task 2: Disassembling and assembling of the PC back to working condition.

Task 3: Installation of MS windows on the personal computer.

Task 4: Installation of Linux on the computer. Dual boot (VMWare) with both Windows and Linux.

Task 5: Installation of BOSS on the computer configure as dual boot (VMWare) with both Windows and BOSS.

Internet & World Wide Web

Task1: Study of different types of Network cables and implement the Cross-Wired Cable and Straight through Cable using Clamping tool. Identify Network Devices and Study basic network commands in Detail.

Task2: Orientation & Connectivity Boot Camp:

a) Configuration of the TCP/IP setting.
b) Demonstration to access the websites and email.
c) Simulation of the WWW on the LAN.

Task 3: Web Browsers:

a) Surfing the web
b) Customization of web browsers with the LAN proxy settings, bookmarks, search toolbars and pop up blockers.
c) Configuration of plug-ins like Macromedia Flash and JRE for applets.

Task 4: Demonstration of using Search Engines & Netiquette

Task 5: Cyber Hygiene:

a) Configuration of computers to be safe on the internet.
b) Customization of browsers to block pop ups, block active x downloads to avoid viruses and/or worms

Task 6: Email Creation: Students should know what how to send e-mails. A few documents/content/different user e-mails would be given to the students for which they need to send – emails. This should be demonstrated to the instructors by the student.

.

LaTeX and WORD

Task 1 – Orientation of La TeX and Microsoft (MS) office or equivalent (FOSS) tool word

Task 2: Using La TeX and Word to create a project certificate. Features to be covered:- Formatting Fonts in word, Drop Cap in word, Applying Text effects, Using Character Spacing, Borders and Colors, Inserting Header and Footer, Using Date and Time option in both La TeX and Word.

Task 3: Creating project abstract Features to be covered:-Formatting Styles, Inserting table, Bullets and Numbering, Changing Text Direction, Cell alignment, Footnote, Hyperlink, Symbols, Spell Check, Track Changes.

Task 4: Creating a Newsletter: Features to be covered:- Table of Content, Newspaper columns, Images from files and clipart, Drawing toolbar and Word Art, Formatting Images, Textboxes, Paragraphs and Mail Merge in word.

EXCEL Excel Orientation:

Task 1: Orientation of MS office or equivalent (FOSS) tool Excel as a Spreadsheet tool.

Task 2: Creating a Scheduler - Features to be covered: Gridlines, Format Cells, Summation, auto fill, Formatting Text

	Task 3: Calculating GPA -. Features to be covered:- Cell Referencing, Formulae in excel – average, std. deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function **LOOKUP/VLOOKUP** **Task 4:** Split cells, freeze panes, group and outline, Sorting, Boolean and logical operators, Conditional formatting. **POWER POINT** **Task 1:** Orientation on basic power point presentations. PPT Orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in PowerPoint. **Task 2:** Interactive presentations - Hyperlinks, Inserting –Images, Clip Art, Audio, Video, Objects, Tables and Charts. **Task 3:** Master Layouts (slide, template, and notes), Types of views (basic, presentation, slide slotter, notes etc), and Inserting – Background, textures, Design Templates, Hidden slides. **AI TOOLS – ChatGPT** **Task 1:** Prompt Engineering: Experiment with different types of prompts to see how the model responds. Try asking questions, starting conversations, or even providing incomplete sentences to see how the model completes them. • Ex: Prompt: "You are a knowledgeable AI. Please answer the following question: What is the capital of France?" **Task 2:** Creative Writing: Use the model as a writing assistant. Provide the beginning of a story or a description of a scene, and let the model generate the rest of the content. This can be a fun way to brainstorm creative ideas • **Ex:** Prompt: "In a world where gravity suddenly stopped working, people started floating upwards. Write a story about how society adapted to this new reality." **Task 3:** Language Translation: Experiment with translation tasks by providing a sentence in one language and asking the model to translate it into another language. Compare the output to see how accurate and fluent the translations are. • Ex:Prompt: "Translate the following English sentence to French: 'Hello, how are you doing today?'"
Text Books and References	**Reference Books:** 1. Comdex Information Technology course tool kit, Vikas Gupta, WILEY Dream tech, 2003 2. The Complete Computer upgrade and repair book, Cheryl A Schmidt, WILEY Dream tech, 2013, 3rd edition 3. Introduction to Information Technology, ITL Education Solutions limited,

	Pearson Education, 2012, 2nd edition 4. PC Hardware - A Handbook, Kate J. Chase, PHI (Microsoft) 5. LaTeX Companion, Leslie Lamport, PHI/Pearson. 6. IT Essentials PC Hardware and Software Companion Guide, David Anfins on and Ken Quamme. – CISCO Press, Pearson Education, 3 rd edition 7. 7. IT Essentials PC Hardware and Software Labs and Study Guide, Patrick Regan– CISCO Press, Pearson Education, 3 rd edition

23CS1201 – DATA STRUCTURES

(Common to CSE, IT & allied branches)

Course Category:	Professional core	**Credits:**	3
Course Type:	Theory	**Lecture - Tutorial - Practical:**	3-0-0
Prerequisite:	Knowledge in programming languages.	**Sessional Evaluation:** **Univ. Exam Evaluation:** **Total Marks:**	30 70 100
Objectives	• To provide the knowledge of basic data structures and their implementations. • To understand importance of data structures in context of writing efficient programs. • To develop skills to apply appropriate data structures in problem solving.		

Course Outcomes	At the end of the course, Student will be able to	
	CO1	Explain the role of linear data structures in organizing and accessing data efficiently.
	CO2	Design, implement, and apply linked lists for dynamic data storage, demonstrating understanding of memory allocation.
	CO3	Develop programs using stacks and queues to handle recursive algorithms, manage program states, and solve related problems.
	CO4	Devise solutions to small scale programming challenges involving data structures such as Trees, Binary search trees and Height-balanced trees.
	CO5	Describe the fundamentals of graphs and recognize scenarios where hashing is an advantage, and design hash-based solutions for specific problems.

Course Content

UNIT - I

Introduction to Linear Data Structures: Definition and importance of linear data structures, Abstract data types (ADTs) and their implementation, Overview of time and space complexity analysis for linear data structures.
Searching Techniques: Linear & Binary Search.
Sorting Techniques: Bubble sort, Selection sort, Insertion Sort.

UNIT - II

Linked Lists: Singly linked lists: representation and operations, doubly linked lists and circular linked lists, Comparing arrays and linked lists, Applications of linked lists.

UNIT - III

Stacks: Introduction to stacks: properties and operations, implementing stacks using arrays and linked lists, Applications of stacks in expression evaluation, backtracking, reversing list etc.
Queues: Introduction to queues: properties and operations, implementing queues using arrays and linked lists, Applications of queues in breadth-first search, scheduling, etc.
Deques: Introduction to deques (double-ended queues), Operations on deques and their applications.

	UNIT - IV **Trees**: Introduction, Types and basic properties, **Binary trees**: definition, Tree traversals, Tree representations, **Binary Search Trees**: Insertion, Deletion, and applications, Introduction to Height-balanced trees, **Heap** – Introduction, Heap data structure types and Heap sort. **UNIT - V** **Graphs:** Introduction, Basic terminologies, Representations, Bi-connected components, Topological sorting. **Hashing:** Brief introduction to hashing and hash functions, Collision resolution techniques: chaining and open addressing, Hash tables: basic implementation and operations, Applications of hashing in unique identifier generation, caching, etc.
Text Books and References	**Text Books:** 1. Data Structures, Algorithms, and Software Principles in C, Thomas A Standish, Addison-Wesley Publishing Company, 1995. 2. Data Structures and algorithm analysis in C, Mark Allen Weiss, Pearson, 2nd Edition. 3. Fundamentals of data structures in C, Ellis Horowitz, Sartaj Sahni, Susan AndersonFreed, Silicon Press, 2008 **Reference Books:** 1. Algorithms and Data Structures: The Basic Toolbox by Kurt Mehlhorn and Peter Sanders 2. C Data Structures and Algorithms by Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft 3. Problem Solving with Algorithms and Data Structures" by Brad Miller and David Ranum 4. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein 5. 5. Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms" by Robert Sedgewick

23CS12P1 – DATA STRUCTURES LABORATORY

(Common to CSE, IT & allied branches)

Course Category:	Professional Core	**Credits:**	1.5
Course Type:	Practical	**Lecture - Tutorial - Practical:**	0-0-3
Prerequisite:	Knowledge in programming languages.	**Sessional Evaluation:** **Univ. Exam Evaluation:** **Total Marks:**	30 70 100
Objectives	The course aims to strengthen the ability of the students to identify and apply the suitable data structure for the given real-world problem. It enables them to gain knowledge in practical applications of data structures.		

Course Outcomes	At the end of the course, Student will be able to	
	CO1	Explain the role of linear data structures in organizing and accessing data efficiently.
	CO2	Design, implement, and apply linked lists for dynamic data storage, demonstrating understanding of memory allocation.
	CO3	Develop programs using stacks and queues to handle recursive algorithms, manage program states, and solve related problems.
	CO4	Devise solutions to small scale programming challenges involving data structures such as Trees, Binary search trees and Height-balanced trees.
	CO5	Describe the fundamentals of graphs and recognize scenarios where hashing is an advantage, and design hash-based solutions for specific problems.

Course Content

LIST OF EXPERIMENTS

Exercise 1: Example programs on **Array Manipulation**

Exercise 2: Implementation of **Linked List** Operations

Exercise 3: Implementation of any two applications on **Linked Lists.**

Exercise 4: Implementation of **Double Linked List** operations and applications.

Exercise 5: Implementation of **Stack Operations** using Arrays and Linked Lists**.**

Exercise 6: Implementation of **Queue Operations** using Arrays and Linked Lists.

Exercise 7: Implementation of **Stack and Queue Applications**

Exercise 8: Implementation of **Binary Search Tree** using Linked List**.**

Exercise 9: Implementation of **Hash table** with collision resolution techniques**.**

	Exercise 10: Implementation of **Simple Cache** using hashing**.**
Text Books and References	**Text Books:** 1. Data Structures, Algorithms, and Software Principles in C, Thomas A Standish, Addison-Wesley Publishing Company, 1995. 2. Data Structures and algorithm analysis in C, Mark Allen Weiss, Pearson, 2nd Edition. 3. Fundamentals of data structures in C, Ellis Horowitz, Sartaj Sahni, Susan AndersonFreed, Silicon Press, 2008 **Reference Books:** 1. Algorithms and Data Structures: The Basic Toolbox by Kurt Mehlhorn and Peter Sanders 2. C Data Structures and Algorithms by Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft 3. Problem Solving with Algorithms and Data Structures" by Brad Miller and David Ranum 4. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein 5. Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms by Robert Sedgewick