



Problem Solving Using Computers| CS 1002 | 3 Credits | 3 0 0 3

Session: January 2024 – May 2024 | Faculty: Dr. Gautam Kumar (Course Coordinator) | Dr. Rajat Goel | Dr. Veena Khandelwal | Mr. Anurag Bhatnagar | Ms. Vineeta Soni | Dr. Shweta Sharma | Mr. Suman Saurabh Sarkar | Dr. Siddhanta Kumar Singh | Dr. Jeya Krishnan.V | Dr. Krati Dubey | Ms. Babita Tiwari | Ms. Bhawna Sharma | Dr. Sandeep Singh | Dr. Amit Kumar Sharma | Dr. Harish Kumar Shakya

Class: B. Tech 1st Year

- A. Introduction:** Problem solving using computers course focuses on basic computer fundamentals, number system and programming in C fundamentals. By means of C language students learn to write a set of instructions to create a program so that desired output can be generated by computer.
- B. Course Outcomes:** At the end of the course, students will be able to
- [CS1002.1]. Understand algorithm development, elements of programming languages and designing flow charts. [CS1002.2]. Apply the basic programming concepts such as tokens, data types, operators, and control statements for implementing programs.
 - [CS1002.3]. Describe and analyze the concepts of array data type (1D and 2D), functions, structure, and union. [CS1002.4]. Illustrate the concept of pointers and file handling.
 - [CS1002.5]. Creating algorithms or pseudo code to solve real life problems using programming constructs.

These course outcomes ensure that students gain a comprehensive understanding of computer programming principles, data manipulation, and problem-solving techniques, while also mastering the use of essential programming constructs and libraries.

C. PROGRAM OUTCOMES

- [PO.1]. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- [PO.2]. Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences.
- [PO.3]. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- [PO.4]. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- [PO.5]. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- [PO.6]. The engineer and society:** Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal, and cultural issues, and the consequent responsibilities relevant to professional engineering practice.
- [PO.7]. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- [PO.8]. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering

practices.

[PO.9]. **Individual and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

[PO.10]. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

[PO.11]. **Project management and finance:** Demonstrate knowledge and understanding of engineering.

and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

[PO.12]. **Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

D. Assessment Plan:

Criteria	Description	Maximum Marks
Internal Assessment (Summative)	MTE I (Closed Book)	30
	Quizzes (Best 3 of 4: 15%) and Assignments (Two: 5%)	30
End Term Exam (Summative)	End Term Exam (Closed Book)	40
	Total	100
Attendance (Formative)	A minimum of 75% Attendance is required to be maintained by a student to be qualified for taking the End Semester examination. The allowance of 25% includes all types of leaves including medical. leaves.	

E. SYLLABUS

Module	CONTENTS
1	Introduction to Computer Programming and Computer Architecture: Algorithms and flowcharts, the von Neumann architecture, programs, assembly language, high level programming languages
2	Fundamentals of Data Types and Variables in Programming: Data types, variables, Enumerated datatypes, Storage Class
3	Understanding Operators, Expressions, and Control Statements: Operators, expressions, Control Statements
4	Working with Arrays and Pointers in Programming: Arrays and pointers
5	Functions, Recursion, and Structures in Programming: Function, recursion, Structures
6	File Handling, Standard Library Functions, and Elementary Data Structures: File Handling, some standard library functions, and some elementary data structures.

F. Text Books

T1. E. Balagurusamy, "Programming in ANSI C", 7th Edition, McGraw Hill Publication, 2017. T2. Y. P. Kanetkar, "Let us C", 16th Edition, BPB Publication, 2017.

G. Reference Books

R1. B. W. Kernighan, D. M. Ritchie, "The C Programming Language", 2nd Edition, Prentice Hall of India, 2014.

R1. B. Gottfried, "Schaum's Outline Series: Programming with C", 3rd Edition, McGraw Hill Publication, 2012.

H. Lecture Plan:

Lecture Number	Topics	Session Outcome	Mode of Delivery	Corresponding CO	Mode of Assessing CO
1	von Neumann architecture, programs, assembly language, high level programming languages	To acquaint basic knowledge of computer system.	Lecture	1002.1	Mid Term, Quiz & End Term
2	Algorithm and Flowchart	To learn process to solve any complex problem	Lecture	1002.1	Mid Term, Quiz & End Term
3.	Data types(table including range, memory, and format specifier), Operators: arithmetic, relational, logical, assignment.	Implementation of various data type	Lecture	1002.2	Mid Term, Quiz & End Term
4.	Storage Class, Enumerated datatype	To gain knowledge of storage class and enumerated data types	Lecture	1002.2	Mid Term, Quiz & End Term
7.	Bitwise, conditional, type-cast, size of, comma, Operator precedence	Implementation of various arithmetic operations	Lecture, Activity (TSP)	1002.2	Mid Term, Quiz & End Term
8.	Operator precedence and associativity, type conversion	Implementation of precedence in programming	Lecture	1002.2	Mid Term, Quiz & End Term
9.	Input and output statements (formatted and unformatted): printf, scanf, gets, puts, getchar, putchar	Implementation of input and output statements	Lecture	1002.2	Mid Term, Quiz & End Term
10.	Decision statements: if, if-else, nested if-else, if-else ladder	Implementation of decision statements	Lecture	1002.3	Mid Term, Quiz & End Term
11.	Switch, break statement	Learning the implementation of switch and break	Lecture	1002.3	Mid Term, Quiz & End Term
12.	Repetitive structures: for, while loops	Learning the implementation of looping	Lecture	1002.3	Mid Term, Quiz & End Term
13.	Repetitive structures: do-while loops, nested loops	Learning the implementation of looping	Lecture	1002.3	Mid Term, Quiz & End Term
14.	More examples on Nested loops of various types	Learning the implementation of looping	Activity (Think Pair Share)	1002.3	Mid Term, Quiz & End Term
15.	Continue and break statements	Describe the usage of continue and break	Lecture	1002.3	Mid Term, Quiz & End Term

16.	1-D array: definition, declaration, initialization, input array, output array	Describe and define array of various data type	Lecture	1002.4	Mid Term, Quiz & End Term
17.	1-D array: Examples of array operations	Describe and define array of various data type	Lecture	1002.4	Mid Term, Quiz & End Term
18.	1-D character array: character array, string, standard functions for string manipulation	Describe and define array of various data type	Lecture	1002.4	Mid Term, Quiz & End Term
19.	1-D character array: user defined functions of standard string manipulation functions	Describe and define array of various data type	Lecture	1002.4	Mid Term, Quiz & End Term
20.	Application of 1-D array: Linear Search and Bubble Sort	Describe use of linear array	Activity	1002.4	Mid Term, Quiz & End Term
21.	Revision – 1	Revision of earlier concepts	Lecture	1002.1 to 1002.4	Mid Term, Quiz & End Term
22.	2-D array: definition, declaration, initialization, input array, output array, one simple program	Describe and define array of various data type	Lecture	1002.4	Mid Term, Quiz & End Term
23.	2-D array: More examples on 2-D arrays	Describe and define array of various data type	Lecture	1002.4	Mid Term, Quiz & End Term
24.	2-D array: More examples on 2-D array	Describe and define array of various data type	Lecture, Activity	1002.4	Mid Term, Quiz & End Term
25.	Pointers: Introduction	Describe functionality of pointers in programming	Lecture	1002.4	Mid Term, Quiz & End Term
26.	1-D Array and pointer	Implementation of 1D array with pointer	Lecture	1002.4	Mid Term, Quiz & End Term
27.	More examples on pointer and operations using pointers	To acquaint more knowledge about pointers	Lecture	1002.4	Mid Term, Quiz & End Term
28.	Some more examples on pointer and operations using pointers	To acquaint more knowledge about pointers	Lecture	1002.4	Mid Term, Quiz & End Term
29.	Functions: introduction to functions, function prototype, call, definition	Describe importance of function and modular programming	Lecture, Activity	1002.4	Mid Term, Quiz & End Term
30.	Call by value and Call by reference	Describe importance of function and modular programming	Lecture	1002.4	Mid Term, Quiz & End Term
31.	More examples using functions	Describe importance of function and modular programming	Lecture	1002.4	Mid Term, Quiz & End Term
32.	Examples using functions continued	Describe importance of function and modular programming	Lecture	1002.4	Mid Term, Quiz & End Term
33.	Revision – 2	Revision of earlier concepts	Lecture	1002.4	Mid Term, Quiz & End Term

34.	Storage classes	Describe usage of storage classes	Lecture	1002.4	Mid Term, Quiz & End Term
35.	Structures: definition, declaration, initialization	Describe usage of structures	Lecture	1002.4	Mid Term, Quiz & End Term
36.	Structures: array of structures	Describe usage of structures	Lecture	1002.4	Mid Term, Quiz & End Term
37.	Union, difference between union and structures	Describe usage of union	Lecture	1002.4	Mid Term, Quiz & End Term
38.	File handling: introduction, operations on files, opening modes	Describe usage of file handling with various operations and modes	Lecture	1002.5	Mid Term, Quiz & End Term
39.	File handling function	Describe usage of file handling with various operations and modes	Lecture	1002.5	Mid Term, Quiz & End Term
40.	File handling function	Describe usage of file handling with various operations and modes	Lecture	1002.5	Mid Term, Quiz & End Term
41.	Revision – 3	Summarize entire syllabus	Lecture	1002.1 1002.2 1002.3	Mid Term, Quiz & End Term
42.	Revision – 4	Summarize entire syllabus	Lecture	1002.4 1002.5	Mid Term, Quiz & End Term

I. Course Articulation Matrix: (Mapping of COs with POs)

CO	STATEMENT	CORRELATION WITH PROGRAM OUTCOMES												CORRELATION WITH PROGRAM SPECIFIC OUTCOMES		
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CS 1002.1	Understand algorithm development, elements of programming languages and designing flow charts.	2	1										1	2	1	1
CS 1002.2	Apply the basic programming concepts such as tokens, data types, operators and control statements for implementing programs.	2		1									2	1		
CS 1002.3	Describe and analyze the concepts of array data type (1D and 2D), functions, structure and union.	3	1										2	1		
CS 1002.4	Illustrate the concept of pointers and file handling	2		2									1			
CS 1002.5	Creating algorithms or pseudo code to solve real life problems using programming constructs	3	2	1									3	3	1	1

1- Low Correlation; 2- Moderate Correlation; 3- Substantial Correlation