Name of Institute: Indus Institute of Technology & Engineering
Name of Faculty: Dr Kaushal Jani

Course code: CE0216
Course name: Programming for Problem Solving
Pre-requisites: -
Student must have basic understanding of Computer Programming Terminology.

Credit points: 3
Offered Semester: I/II – All Branch

Course coordinator
Full name: Dr Kaushal Jani
Department with siting location:CSE Department,B-401 lab,Bhanwar Building
Telephone: N/A
Email: kaushaljani.ce@indusuni.ac.in
Consultation times:
Saturday 9:30 am to 4:00 pm

Course lecturer
Full name: Dr Kaushal Jani
Department with siting location:CSE Department,B-401 lab,Bhanwar Building
Telephone: N/A
Email: kaushaljani.ce@indusuni.ac.in
Consultation times:
Saturday 9:30 am to 4:00 pm

Full name: Asst. Prof. Zalak Trivedi
Department with sitting location:4th Floor,CSE Department,Bhanwar Building
Telephone: N/A
Email: cse.hod@indusuni.ac.in
Consultation times:
Saturday 9:30 am to 4:00 pm

Full name: Asst. Prof. Chirag Bhatt
Department with sitting location:4th Floor,CSE Department,Bhanwar Building
Telephone:
Email: chiragbhatt.ce@indusuni.ac.in
Consultation times:
Saturday 9:30 am to 4:00 pm

Students will be contacted throughout the session via mail with important information relating to this course.

Course Objectives

1) To familiarize the student with basic concepts of computer programming and developer tools.
2) To describe the parts of the computer system.
3) To describe functioning of computer components.
4) To describe the process of problem-solving using computer

5) To describe the design an algorithmic solution for a given problem
6) To describe a writing method for maintainable C program for a given algorithm.
7) To describe the importance of C program for simple applications of real-life using structures and files.
8) The students will be able to enhance their analyzing and problem-solving skills and use the same for writing programs in C.

Course Outcomes (CO)

By participating in and understanding all facets of this course a student will be able to:

i. To *learn* broad perspective about the uses of computers in engineering industry.

ii. To *develop* the ability to analyze a problem, develop an algorithm to solve it.

iii. To *develop* basic understanding of computers, the concept of algorithm and algorithmic thinking.

iv. To *apply* basic programming principles using C language.

v. To *apply* basic C program structure in software development

vi. To *determine* fundamental principles of problem solving in software engineering through various programming languages

Course Outline

UNIT-I	[8 hours]
Introduction to Programming	
What is programming? Problem solving methods with Examples-Algorithm and Flowchart, Types of Programming languages, Characteristics of higher-level language, Some Programming languages Introduction to 'C' Introduction, Importance of C, Sample C programs, Basic structure of C programs, programming style, executing a C program. Introduction, Character Set, C tokens, Keywords and Identifiers, Constants, Variables, Data types, Declaration of Variables, Defining symbolic constants	
Operators and Expression	
Introduction, Arithmetic of Operators, Relational Operators, Logical Operators, Assignment Operators, Increment and Decrement Operators, Conditional Operators, Bitwise Operators, Special Operators, Arithmetic Expressions, Evaluation of expressions, Precedence of arithmetic operators, Type conversions in expressions, Mathematical function	
UNIT-II	[8 hours]
Decision Making Statements	
Introduction, Decision making with IF statement, Simple IF statement, the IF ELSE statement, Nesting of IF … ELSE statements, The ELSE IF ladder, The switch statement, the turnery (? :) Operator, the GOTO statement Looping WHILE statement, the DO statement, The FOR statement, jumps in loops Break and continue Array & Handling of Character strings: Introduction, One-dimensional arrays, Two-dimensional arrays, Initialization of two-dimensional arrays, Concept of Multidimensional arrays	
Looping	
WHILE statement, the DO statement, The FOR statement, jumps in loops Break and continue	
Array & Handling of Character strings	

Introduction, One-dimensional arrays, Two-dimensional arrays, Initialization of two-dimensional arrays, Concept of Multidimensional arrays		

UNIT-III	**[8 hours]**
Handling of Character strings	

Introduction, Declaring and initializing string variables, reading string from terminal, writing string to screen, Arithmetic operations on characters, Putting string together, String Operations: String Copy, String Compare, String Concatenation and String Length, String Handling functions, Table of strings

User-Defined Functions

Introduction, need for user-defined functions, return values and their types, calling a function, category of functions, no arguments and no return values, Arguments with return values, Handling of non-integer functions, Nesting of functions, Recursion, Functions with arrays, The scope and Lifetime of variables in functions

UNIT-IV	**[8 hours]**
Pointers	

Introduction, understanding pointers, Accessing the address of variable, Declaring and initializing pointers, accessing a variable through its pointer, Pointer expressions, Pointer increments and scale factor, Pointers and arrays, Pointers and character strings, Pointers and Functions, Pointers and structures

Structures and Unions

Introduction, Structure definition, Giving values to members, Structure initialization, Comparison of structures, Arrays of structures, Arrays within structures, Structures within Structures, Structures and functions, Union

Introduction to Object Oriented Concepts & Programming

Review of fundamental concepts of Object-oriented programming, Introduction to C++, class and objects, Functions in C++, Constructors & Destructor

Method of delivery
Chalk and Board, PowerPoint presentation, Model generation, demonstration of devices, cables

Study time
3 hours theory, 2 hours practical

CO-PO Mapping (PO: Program Outcomes)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	-	-	1	-	-	-	-	-	-	1	-
CO2	1	1	-	-	-	-	-	-	-	-	1	-
CO3	2	-	-	2	-	-	-	-	-	-	-	-
CO4	2	-	-	3	-	-	-	-	-	-	-	-
CO5	3	3	-	-	-	-	-	-	-	-	-	-
CO6	3	3	-	-	2	-	1	-	-	-	-	-

Blooms Taxonomy and Knowledge retention (For reference)
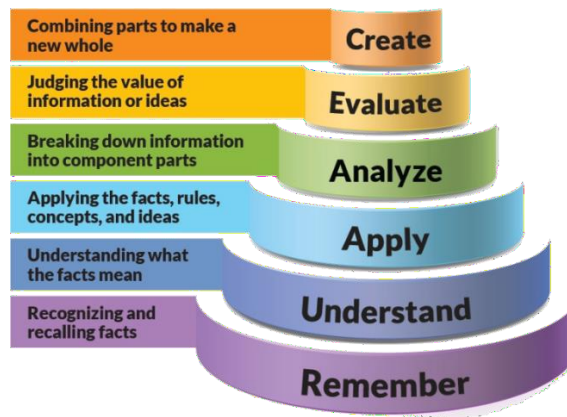(Blooms taxonomy has been given for reference)
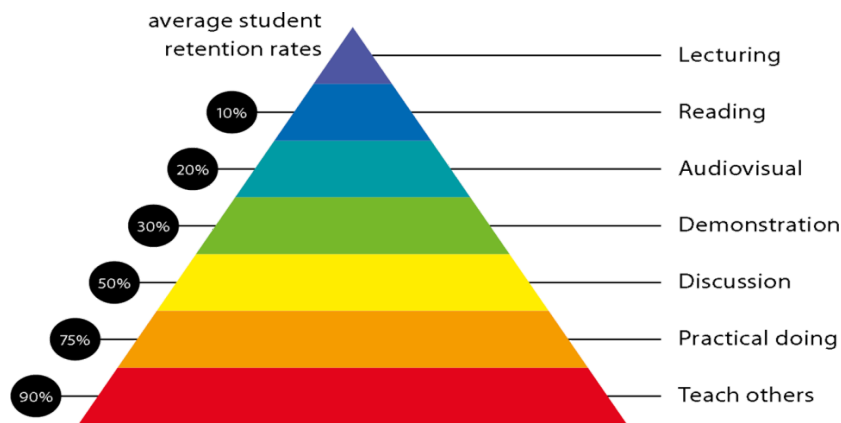
Figure 1: Blooms Taxonomy

Figure 2: Knowledge retention

Graduate Qualities and Capabilities covered

(Qualities graduates harness crediting this Course)

General Graduate Qualities	Specific Department of_____ Graduate Capabilities
Informed Have a sound knowledge of an area of study or profession and understand its current issues, locally and internationally. Know how to apply this knowledge. Understand how an area of study has developed and how it relates to other areas.	**1 Professional knowledge, grounding & awareness**
Independent learners Engage with new ideas and ways of thinking and critically analyze issues. Seek to extend knowledge through ongoing research, enquiry and reflection. Find and evaluate information, using a variety of sources and technologies. Acknowledge the work and ideas of others.	**2 Information literacy, gathering & processing**
Problem solvers Take on challenges and opportunities. Apply creative, logical and critical thinking skills to respond effectively. Make and implement	**4 Problem solving skills**

decisions. Be flexible, thorough, innovative and aim for high standards.		
Effective communicators Articulate ideas and convey them effectively using a range of media. Work collaboratively and engage with people in different settings. Recognize how culture can shape communication.	**5 Written communication**
	6 Oral communication
	7 Teamwork
Responsible Understand how decisions can affect others and make ethically informed choices. Appreciate and respect diversity. Act with integrity as part of local, national, global and professional communities.	**10 Sustainability, societal & environmental impact**

Practical work:

UNIT	TOPIC/DEFINITION		HOURS
I	**Using input and output statements, Operators**		**8**
	1	Write a program to print the address of INDUS.	
	2	Write a program to perform basic arithmetic operators on given two numbers.	
	3	Find the area and perimeter of square and rectangle and circle. Input the side(s) through the keyboard. (use PIE as symbolic constant)	
	4	Write a program to swap values of 2 variables (i) with extra variable and (ii) without using an extra variable.	
	5	Write a program to print the ASCII value of a given character.	
	6	Write a program to enter the integer number and convert it into Rs and Paisa.	
	7	Write a program to enter two numbers. Make the comparison between them with conditional operator. If the first number is greater than second perform multiplication otherwise division operation.	
	8	Write a program to enter the temperature in Fahrenheit and convert it to Celsius.[C = ((F-32)*5)/9]	
	9	Write a program to calculate simple interest.	
	10	Write a program to enter marks of five subject of a student and calculate its average, percentage.	
II	**Using conditional statements**		**8**
	1	Write a program to find the maximum of (i) two integer values and (ii) three integer values.	
	2	Write a program to check whether the given character is a vowel of not.	
	3	Write a program that reads a number from 1 to 7 and accordingly it should display MONDAY to SUNDAY (if- else if).	
	4	Write a menu driven program to perform the arithmetic operations.	

	5	Write a program to print number of days in a given month using switch statement. The program requires month number (between 1 to 12) as an input and then displays number of days in that month.	
	6	Write a program to check whether a given value is even or odd.	
	7	Write a program to calculate total salary of an employee.	

		total salary = basic + da + hra + ta. da = 50% of basic.	
		Basic **hra** **ta** <6000 400 100 6001>= & <10000 1400 300 >=10000 2400 700	
III		**Using control statements**	**10**
	1	Write a program to print 1 to 10 numbers using while loop.	
	2	Write a program to read any 7 numbers and print the average value using for loop.	
	3	Write a program to reverse a given integer number.	
	4	Write a program to print Fibonacci series of given number.	
	5	Write a program to find factorial of a number.	
	6	Write a program to check whether a number is a Krishnamurthy number or not. Krishnamurthy number is one whose sum of factorial of digits equals the number. Example: 145 1! + 4! + 5! = 1 + 24 + 120 = 145	
	7	Write a program to check whether the number is Armstrong or not. Example:153----$1^3 + 5^3 + 3^3$ = 1 + 125 + 27 153	
	8	Write a program to list all prime numbers within given range.	
	9	Write a program to draw following patterns: * 1 5 4 3 2 1 A ** a b 4 3 2 1 A B *** 1 2 3 3 2 1 A B C **** a b c d 2 1 A B C D ***** 1 2 3 4 5 1 A B C D E 1 1 121 0 1 12321 1 0 1 1234321 0 1 0 1	
IV		**Array And Strings**	**10**
	1	Write a program to read 10 integers in an array. Find the addition of all elements.	
	2	Write a program to find number of odd and even elements from the 1- D array.	
	3	Write a program to reverse the elements of array and store it in another array.	
	4	Write a program to sort elements of array.	
	5	Write a Program to print Addition of two matrices.	

	6	Program to remove duplicate numbers from a list of numbers and print the list without duplicate numbers.	
	7	Write a Program to print Multiplication of two matrices.	
	8	Read the marks of five subjects obtained by five students in an examination. Display the top two student's codes and their marks.	
	9	Write a program to insert an element in an array at specified position.	
	10	Write a program to find the length of a string.	
	11	Write a program to reverse the string.(without inbuilt Function)	
	12	Write a program to convert a string in to lower case and upper case.	
	13	Write a menu driven program for the implementation of all build-in string functions.	
	14	Program to extract n characters starting from m in a given string. (String, n and m should be provided as inputs).	
	15	Find out occurrence of each character in a given string.	
V		**Structure & Union**	4
	1	Write a program to define structure with tag state with fields state name, number of districts and total population. Read and display the data.	
	2	Write a program to create a structure of 5 student's roll_no and name and display the records. Use array of structure	
	3	Write a program to create structure of bank with accno, holder_name and balance and display them for n holders whose balance is less than 5000.	
	4	Write a program to create union of student's roll_no and name and display the records.	
VI		**Pointers & Functions**	10
	1	Write a program that demonstrates the use of address of (&) and pointer (*) operator.	
	2	Write a program to read and display values of an integer array. Allocate space dynamically for the array.	
	3	Write a program to display the content of 1-D array using pointer.	
	4	Write a program to sum given two integer numbers using function. .	
	5	Write a program using function to count the area of circle, triangle, rectangle and square.	
	6	Write a program using user defined function even _odd. With argument and check whether the no is even or odd.	
	7	Write a program using function with array, takes input of five subject's marks and count the percentage and display result.	
	8	Write a function which accepts a character array as argument from the user. The function should convert all the lowercase characters into uppercase case	
	9	Write a function using pointer parameter that calculate maximum element from given array of integer number.	

	10	Write a program that demonstrates call by value and call by reference concept in function argument.	

Lecture/tutorial times

(Give lecture times in the format below)

Lecture	Day	00.00pm – 00.00 am/pm	Room No

Attendance Requirements

The University norms states that it is the responsibility of students to attend all lectures, tutorials, seminars and practical work as stipulated in the course outline. Minimum attendance requirement as per university norms is compulsory for being eligible for semester examinations.

Text books

1. Programming in ANSI C by Balagurusamy, publisher: TMH

Reference Books:

1. Introduction to C by Reema Thareja, Publisher-Oxford
2. Programming with ANSI and Turbo C, by Ashok N Kamthane, Publisher – Pearson Education.
3. Let us C, by Yashwant Kanitkar, Publisher – BPB Publication

Additional Materials
Web Resource http://nptel.ac.in/courses/106105085/2
https://onlinecourses.nptel.ac.in/iitk_cs_101/preview
https://onlinecourses.nptel.ac.in/noc15_cs15/preview

ASSESSMENT GUIDELINES
Your final course mark will be calculated from the following:

CIE-Theory (60 Marks)	CIE-Practical (60 Marks)
Mid Sem Exam: 40 Marks Assignment :15 Marks Attendance : 5 Marks	Practical Performance + Regularity – 10 Marks Practical Test - 30 Marks Practical File - 10 Marks
ESE-Theory- 100 Marks	ESE-Practical-40 Marks
Total: 200 Marks	

SUPPLEMENTARY ASSESSMENT

Students who receive an overall mark less than 40% in internal component or less than 40% in the end semester will be considered for supplementary assessment in the respective components (i.e internal component or end semester) of semester concerned. Students must make themselves available during the supplementary examination period to take up the respective components (internal component or end semester) and need to obtain the required minimum 40% marks to clear the concerned components.

Practical Work Report/Laboratory Report:

A report on the practical work is due the subsequent week after completion of the class by each group.

Late Work

Late assignments will not be accepted without supporting documentation. Late submission of the reports will result in a deduction of -% of the maximum mark per calendar day

Format

All assignments must be presented in a neat, legible format with all information sources correctly referenced. **Assignment material handed in throughout the session that is not neat and legible will not be marked and will be returned to the student.**

Retention of Written Work

Written assessment work will be retained by the Course coordinator/lecturer for two weeks after marking to be collected by the students.

University and Faculty Policies

Students should make themselves aware of the University and/or Faculty Policies regarding plagiarism, special consideration, supplementary examinations and other educational issues and student matters.

Plagiarism - Plagiarism is not acceptable and may result in the imposition of severe penalties. Plagiarism is the use of another person's work, or idea, as if it is his or her own - if you have any doubts at all on what constitutes plagiarism, please consult your Course coordinator or lecturer. Plagiarism will be penalized severely.

Do not copy the work of other students.
Do not share your work with other students (except where required for a group activity or assessment.

Course schedule (subject to change)

(Mention quiz, assignment submission, breaks etc as well in the table under the Teaching Learning Activity Column)

	Week #	Topic & contents	CO Addressed	Teaching Learning Activity (TLA)
	Weeks 1	Introduction to programming, algorithms	I	Chalk & Board
	Weeks 2	Basics of C, Operators and Expression	I	Presentation, Chalk & Board
	Week 3	Control structures, Decision making statements, if statement, if..else statement, else..if statement, nested	I	Chalk & Board, Discussion

		if..else, switch statement		
	Week 4	Looping, while, do..while, for loop, goto statement, break and continue statement	II	Presentation & Discussion
	Week 5	Array and handling of character string, one dimensional array, two-dimensional array,	II	Chalk & Board, Presentation
	Week 6	Multidimensional array	II	Chalk & Board
	Week 7	String handling functions	III	Chalk & Board
	Week 8	User defined function, Nesting of function	III	Chalk & Board
	Week 9	Recursion, The scope and lifetime of variables in function	III	Chalk & Board
	Week 10	Understanding pointers, Accessing the address of variable, Declaring and initializing pointers	II,III	Chalk & Board
	Week 11	Accessing a variable through its pointer, Pointer expressions, Pointer increments and scale factor	III	Chalk & Board, Discussion
	Week 12	Structure definition, Giving values to members, Structure initialization, Comparison of structures	IV	Chalk & Board
	Week 13	Arrays of structures, Arrays within structures, Structures within Structures, Structures and functions, Union	IV	Chalk & Board
	Week 14	Review of fundamental concepts of Object-oriented programming	IV, V	Chalk & Board, Discussion
	Week 15	Introduction to C++, class and objects, Functions in C++, Constructors & Destructor	V	Presentation, Chalk & Board

Subject Mind Mapping

	Semester1&2	Semester 3	Semester 4	Semester 5	Semester 6	Semester 7	Semester 8
Programming and Software Development	Principles of Programming	Object Orieneted concepts and Programming / Web Design	Core Java Prog / Object Orieneted Design and Analysis	Web technology	Advanced java Technology / Software Engineering / Python Programming	Android / Web data Management / Advanced .Net	
System tools and Technology		Principle of Database Management System	Operating System		Computer Graphics	Mobile Computing	
System Architecture and design		Digital Logic Design	Computer Organization and Architecture	Microprocessor / Formal language and Automata Theory	System Programming	Compiler Design	
Networking		Data Communication and Engineering		Computer Networks	Cryptography & Network Security		
Mathematics	ICLA / DCMA	DEIT	CANM				
Advanced Learning				Design and Analysis of Algorithms	Parallel Computing Soft Computing Data warehousing Distributed OS	Big Data Iot & Application High Performance Computing	

Project