UPES
UNIVERSITY WITH A PURPOSE

Year : 2020-21
Semester : I

Name of Faculty	: Sunil Kumar	Course Code: **CSEG 1110**
1. Course	: Principles of Programming Languages Lab	L: 0
2. Program	: B.Tech. CSE	T: 0
3. Target	: 50%	P: 1
		C: 1

COURSE PLAN

Target	50% (marks)
Level-1	40% (population)
Level-2	50% (population)
Level-3	60% (population)

1. Method of Evaluation

UG ✔	PG
Quizzes/Tests, Assignments (30%)	Quizzes/Tests, Assignments, seminar (50%)
Mid Examination (20%)	End semester (50%)
End examination (50%)	

2. Passing Criteria

Scale	PG	UG ✔
Out of 10 point scale	SGPA – “6.00” in each semester CGPA – “6.00” Min. Individual Course Grade – “C” Course Grade Point – “4.0”	SGPA – “5.0” in each semester CGPA – “5.0” Min. Individual Course Grade – “C” Course Grade Point – “4.0”

*for PG, passing marks are 40/100 in a paper
*for UG, passing marks are 35/100 in a paper

3. Pedagogy

Solution to the problems should be designed (algorithm/flow-chart/pseudocode) and tested. After obtaining a successful design, the solution to the problem is implemented using C language. Students are evaluated based on Performance (via efficient design, implementation) and record keeping, and preparation of students (via viva-voce/quiz).

4. References:

Text Books	Web resources	Reference books
Yashavant P. Kanetkar, “Let us C”.	Blackboard resources	E.Balagurusamy, “Programming in ANSI C”.

UPES
UNIVERSITY WITH A PURPOSE

Year : 2020-21
Semester : I

Name of Faculty		: Sunil Kumar	Course Code: **CSEG 1110**
1.	Course	: Principles of Programming Languages Lab	L: 0
2.	Program	: B.Tech. CSE	T: 0
3.	Target	: 50%	P: 1
			C: 1

GUIDELINES TO STUDY THE SUBJECT

Instructions to Students:

1. Go through the 'Syllabus' in the Black Board section of the web-site(https://learn.upes.ac.in) in order to find out the Reading List.
2. Get your schedule and try to pace your studies as close to the timeline as possible.
3. Get your on-line lecture notes (Content, videos) at Lecture Notes section. These are our lecture notes. Make sure you use them during this course.
4. Check your blackboard regularly
5. Go through study material
6. Check mails and announcements on blackboard
7. Keep updated with the posts, assignments and examinations which shall be conducted on the blackboard
8. Be regular, so that you do not suffer in any way
9. C**ell Phones and other Electronic Communication Devices:** Cell phones and other electronic communication devices (such as Blackberries/Laptops) are not permitted in classes during Tests or the Mid/Final Examination. Such devices MUST be turned off in the class room.
10. **E-Mail and online learning tool:** Each student in the class should have an e-mail id and a pass word to access the LMS system regularly. Regularly, important information – Date of conducting class tests, guest lectures, via online learning tool. The best way to arrange meetings with us or ask specific questions is by email and prior appointment. All the assignments preferably should be uploaded on online learning tool. Various research papers/reference material will be mailed/uploaded on online learning platform time to time.
11. **Attendance:** Students are required to have minimum attendance of 75% in each subject. Students with less than said percentage shall NOT be allowed to appear in the end semester examination.

This much should be enough to get you organized and on your way to having a great semester! If you need us for anything, send your feedback through e-mail to your concerned faculty. Please use an appropriate subject line to indicate your message details.

There will no doubt be many more activities in the coming weeks. So, to keep up to date with all the latest developments, please keep visiting this website regularly.

Year : 2020-21
Semester : I

Name of Faculty	: Sunil Kumar	Course Code: **CSEG 1110**
1. Course	: Principles of Programming Languages Lab	L: 0
2. Program	: B.Tech. CSE	T: 0
3. Target	: 50%	P: 1
		C: 1

RELATED OUTCOMES

1. The expected outcomes of the Program are:

PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	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.
PO4	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.
PO5	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.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	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.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and team-work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	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.
PO11	Project management and finance: Demonstrate knowledge and understanding of the 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.
PO12	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.

UPES
UNIVERSITY WITH A PURPOSE

Year : 2020-21
Semester : I

Name of Faculty	: Sunil Kumar	Course Code: **CSEG 1110**
1. Course	: Principles of Programming Languages Lab	L: 0
2. Program	: B.Tech. CSE	T: 0
3. Target	: 50%	P: 1
		C: 1

2. The expected outcomes of the Specific Program are:

PSO1	Perform system and application programming using computer system concepts, concepts of Data Structures, algorithm development, problem solving and optimizing techniques.
PSO2	Apply software development and project management methodologies using concepts of front-end and back-end development and emerging technologies and platforms.
PSO3	-

3. The expected outcomes of the Course are:

CO1	Familiarize programming environment and use of apt program control flow in problem solving.
CO2	Develop skills to work with functions, pointers, arrays and searching algorithms.
CO3	Implement structures, unions and dynamic memory allocation.
CO4	Demonstrate file handling and sorting.

4. CO-PO/PSO Relationship Matrix

PO/PSO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	1	2	1	0	0	0	0	0	0	0	1	3	0	-
CO2	1	1	2	1	0	0	0	0	0	0	0	1	3	0	-
CO3	1	1	2	1	0	0	0	0	0	0	0	1	3	0	-
CO4	1	1	2	1	0	0	0	0	0	0	0	1	3	0	-
Average	1	1	2	1	0	0	0	0	0	0	0	1	3	0	-

1- Slight (low) 2- Moderate (Medium) 3-Substantial (high)

5. Course Outcomes assessment plan:

Components Course Outcomes	Experiments*	Quiz/Viva-1	Quiz/Viva-2
CO1	✓	✓	
CO2	✓	✓	
CO3	✓		✓
CO4	✓		✓

*Experimentation & Report

UPES
UNIVERSITY WITH A PURPOSE

Year : 2020-21
Semester : I

Name of Faculty	: Sunil Kumar	Course Code: **CSEG 1110**
1. Course	: Principles of Programming Languages Lab	L: 0
2. Program	: B.Tech. CSE	T: 0
3. Target	: 50%	P: 1
		C: 1

BROAD PLAN OF COURSE COVERAGE

Course Activities:

S.No.	Description	Planned			Remarks
		Batch-1	Batch-2	No. of Sessions	
1.	Linux Commands and C programming Environment			1	CO1
2.	Basics of Problem Solving and Program Control Flows			1	CO1
3.	Programming Sequential Logic			1	CO1
4.	Conditional Branching			1	CO1
5.	Working with Loops/Iterations			1	CO1
6.	Functions, Recursion and Pointers			1	CO2
7.	1D arrays and String handling			1	CO2
8.	2D arrays and Searching techniques			1	CO2
9.	Structure and Union			1	CO3
10.	Dynamic Memory Allocation			1	CO3
11.	Sorting and File Handling			1	CO4

Sessions: Total No. of Instructional periods available for the course

UPES
UNIVERSITY WITH A PURPOSE

Year : 2020-21
Semester : I

Name of Faculty	: Sunil Kumar	Course Code: **CSEG 1110**
1. Course	: Principles of Programming Languages Lab	L: 0
2. Program	: B.Tech. CSE	T: 0
3. Target	: 50%	P: 1
		C: 1

SESSION PLAN

EXPERIMENT -1: Linux Commands and C programming Environment

Objective: To familiarize Linux Commands and demonstrate a clear understanding of the C-programming environment.

List of Lab Activities:

1. Study of Linux Commands
   a. Working with Directories: mkdir, rmdir, dir, pwd, cd, ls
   b. Handling Files: vi, gedit, more, cp, mv, rm
2. Familiarization of the following
   a. Structure of C program [Example: C program to print “My name is …”]
   b. Execution Environment & Stages of Compilation. Use Linux vi editor on the terminal window.

EXPERIMENT-2: Basics of Problem Solving and Program Control Flows

Objective: To demonstrate use of algorithm and flowchart to design solutions for problems with appropriate selection of program control flow.

List of Lab Activities:

Identify the suitable program control flow to solve the given problem. Write algorithm, draw flow chart, prepare test cases and test the design for completeness.

1. Given 2 numbers. Calculate sum, difference, multiplication and division.
2. Find if the given number is even or not.
3. Find the biggest of three numbers.
4. Multiply two numbers without using arithmetic multiplication operator (*).

EXPERIMENT-3: Programming Sequential Logic

Objective: To code sequential logic in C language

List of Lab Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Obtain the required inputs and compute the areas of the following shapes: (i) Parallelogram (with base and height), (ii) Trapezoid (with height, long base, short base), (iii) Rhombus (with height and side), (iv) Sphere (with radius), (v) Ellipse (with major and minor radius)
2. Given two numbers. Demonstrate the swapping of the values (i) using a third variable (ii) without using a third variable
3. Convert temperature from Celsius to Fahrenheit and Kelvin.
4. Print the given days in years-month-days format. E.g. 396 days = 1 year, 1 month, 1 day

UPES
UNIVERSITY WITH A PURPOSE

Year : 2020-21
Semester : I

Name of Faculty	: Sunil Kumar	Course Code: **CSEG 1110**
1. Course	: Principles of Programming Languages Lab	L: 0
2. Program	: B.Tech. CSE	T: 0
3. Target	: 50%	P: 1
		C: 1

EXPERIMENT-4: Conditional Branching

Objective: To understand conditional logic of execution; applying conditional branching structures in C (if, if-else, if-else-if ladder, nested-if, switch-case).

List of Lab Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Find the biggest of 3 numbers.
2. Check whether a given year is leap year or not.
3. Find the roots of a quadratic equation.
4. Check if a given character is a vowel or consonant using Switch-Case statement.

List of Practice Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Find whether a given number is even or odd.
2. Find if the given number is positive, negative or zero.
3. Check if the given number is divisible by 2 and 3 or not.
4. Check whether a given character is an alphabet, digit or special character.

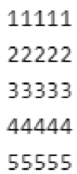
EXPERIMENT-5: Working with Loops/Iterations

Objective: To understand the concepts of Looping with Iterations; applying while, do-while and for constructs.

List of Lab Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Given positive number ‘n’, generate all the Armstrong numbers between 1 and n. [Hint: A 3-digit number (Ex. 153) is an Armstrong number if the sum of cube of each digit (1^3+5^3+3^3) is equal to 153]
2. Multiple two given numbers without using the arithmetic binary multiplication operator using for loop.
3. Find the sum of digits of a number using while loop.
4. Given value of ‘n’, find the sum of the series 1+ 1/2 + 1/3 + 1/4 + 1/5 + …+ 1/n.
5. Print the given pattern using nested for loop.



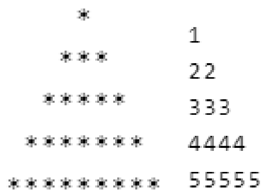
```
11111
22222
33333
44444
55555
```

List of Practice Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Given positive number ‘n’, generate all the Armstrong numbers between 1 and n. [Hint: A 3-digit number (Ex. 153) is an Armstrong number if the sum of cube of each digit (1^3+5^3+3^3) is equal to 153]

2. Generate the first ‘n’ terms of a Fibonacci sequence. [Hint: The first and second terms of a Fibonacci sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence.]
3. Print numbers which are divisible by 3 and 5 from the first ‘n’ natural numbers.
4. Generate all the prime numbers between 1 and n, where n is a value supplied by the user.
5. Reverse the digits of a given number and check if given number is a Palindrome or not using do-while.
6. Using a menu driven control, print the given patterns as per user choice.



```
    *             1
   ***            22
  *****           333
 *******          4444
*********         55555
```

EXPERIMENT-6: Functions, Recursion and Pointers

Objective: To understand the concept of Functions and Looping with Recursion.

List of Lab Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Function main() gets a number and calls the following three functions
   a. “void armstrong(int)” checks if the given number is a Armstrong number or not.
   b. “void coprime(int) reverses the given number and checks if the given number and reversed number are coprime.
   c. “int factorial(int) computes the factorial of the given number using recursion and returns to main().
2. Function main() gets two numbers from the user and calls three functions in the given order:
   a. “int triangle_area(int base, int height)” returns the area of the right-angled triangle to main().
   b. “void swap(int *, int*)” swaps the two numbers using bitwise operator and displays them.
   c. “float* remainder (int a, int b)” returns the remainder of a/b to main().

List of Practice Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Find the maximum and minimum between two given numbers using functions.
2. Write a function to reverse any number using recursion.

EXPERIMENT-7: 1D Arrays & Strings

Objective: To understand the concept of Arrays, manipulating array elements, arrays of unknown or varying size.

List of Lab Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Find sum of all array elements using recursion.
2. Create an array ‘a1’ with ‘n’ elements. Insert an element in i^{th} position of ‘a1’ and also delete an element from j^{th} position of ‘a1’.
3. Convert uppercase string to lowercase using for loop.

List of Practice Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Create an array ‘a1’ with ‘n’ elements.
   a. Copy all elements of ‘a1’ into another array ‘a2’ using pointers. Display the contents of both the arrays using pointers.
   b. Merge the contents of ‘a1’ and ‘a2’ into a new array ‘a3’.
2. Find the total number of alphabets, digits or special characters in a string.
3. Find whether the entered string is palindrome or not.
4. Count the number of words in a string.
5. Find the sum of rows and columns of matrix of given order (row x column).
6. Count how many even numbers are there in a given integer array. [Hint: Linear Search]
7. Find out the largest and smallest number in a given array. [Hint: Linear Search]
8. Find the Transpose and Inverse of a matrix.
9. Find the product of two matrices using pointers.
10. Find if the given matrix of order (m x n) is a Sparse matrix or not. [Assume that a matrix can become a sparse matrix if more than half the total number of its elements have the value zero]
11. Pass an array of ‘n’ integers to a function which sorts them using selection sort algorithm.
12. Store ‘n’ integers in an array. Find the number of comparisons made while searching a number using Linear and Binary Search techniques.

EXPERIMENT-8: 2D Arrays & Searching

Objective: To understand the concept of 2D Arrays and searching techniques.

List of Lab Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Find the sum of rows and columns of matrix of given order.
2. Count how many even numbers are there in a given integer array. [Hint: Linear Search]
3. Store ‘n’ integers in an array in ascending or descending order. Search for a number with binary search technique.

List of Practice Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Find the Transpose and Inverse of a matrix.
2. Find the product of two matrices using pointers.
3. Find if the given matrix of order is a Sparse matrix or not. [Assume that a matrix can become a sparse matrix if more than half the total number of its elements have the value zero]
4. Find out the largest and smallest number in a given array. [Hint: Linear Search]

5. Store ‘n’ integers in an array in ascending or descending order. Find the number of comparisons made while searching for a number when using linear and binary search technique.

EXPERIMENT-9: Structure and Union

Objective: To understand the concept of structure and union.

List of Lab Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Design a structure ‘product’ to store the details of the product purchased like product name, price per unit, number of quantities purchased, and amount spent. Get the name, price per unit, and number of quantities of the product purchased. Calculate the amount spent on the product and then display all the details of the procured product using structure pointers.
2. Design a structure ‘student_record’ to store student details like name, SAP ID, enrollment number, date of registration and data of birth. The element date of joining is defined using another structure ‘date’ to store date details like day, month, and year. Get data of ‘n’ students and then print the entered values [Hint: Use concept of Nested structures and Array of Structures].
3. Design a union ‘product’ to store the details of the product purchased like product name, price per unit, number of quantities purchased, and amount spent. Get the name, price per unit, and number of quantities of the product purchased. Calculate the amount spent on the product and then display all the details of the procured product using union pointers.

List of Practice Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Add two complex numbers by passing structure to a function as argument.

EXPERIMENT-10: Dynamic Memory Allocation

Objective: To understand the concept of dynamic memory allocation.

List of Lab Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Design a structure ‘subject’ to store the details of the subject like subject name and subject code. Using structure pointer allocate memory for the structure dynamically so as to obtain details of ‘n’ subjects using for loop.
2. Use self-referential structure to handle its elements with random and dynamically allocated memory.

List of Practice Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

UPES

UNIVERSITY WITH A PURPOSE

Year : 2020-21
Semester : I

Name of Faculty	: Sunil Kumar	Course Code: **CSEG 1110**
1. Course	: Principles of Programming Languages Lab	L: 0
2. Program	: B.Tech. CSE	T: 0
3. Target	: 50%	P: 1
		C: 1

1. Use self-referential structure with random and dynamically allocated memory to handle its elements in
   a. LIFO format
   b. FIFO format

EXPERIMENT-11: File Handling and Sorting

Objective: To understand the concept of Files, Operations on Files, and working with different sorting algorithms.

List of Lab Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Merge the contents of two files to another file.
2. Create a file with a list of 'n' integers. Read the numbers from the file into an array and sort them in ascending order. [Hint: Bubble Sort algorithm.]

List of Practice Activities:

Write algorithm and C program, compile, execute and test the code using Linux C compiler with suitable test cases.

1. Find a word in a text file and replace it with another word.
2. Input numbers from file using fscanf() and write even, odd and prime numbers to different files.
3. Obtain 'n1' integers into an array (a1) and 'n2' integers into another array (a2). Sort 'a1' with Insertion sort algorithm and 'a2' with Selection sort algorithm. Write the elements of these two sorted arrays into a file in sorted manner.