

B. Tech. Semester I/II

Subject Name: Programming for Problem Solving

Subject Code: BTCO12107

Type of course: Engineering Science

Prerequisite: Zeal to learn the subject

Rationale: To develop understanding of C and Python programming languages. Introduce and build the required skills for problem solving through logical thinking. To achieve proficiency in necessary skills for problem solving using C and Python programming languages.

Teaching and Examination Scheme:

TEACHING SCHEME				Theory Marks			Practical Marks		Total
L	T	P	C	TEE	CA1	CA2	TEP	CA3	
2	0	4	4	60	25	15	30	20	150

CA1: Continuous Assessment (assignments/projects/open book tests/closed book tests **CA2:** Sincerity in attending classes/class tests/ timely submissions of assignments/self-learning attitude/solving advanced problems **TEE:** Term End Examination **TEP:** Term End Practical Exam (Performance and viva on practical skills learned in course) **CA3:** Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course

Content:

Sr. No.	Topics	Teaching Hrs.	Module Weightage
1.	Introduction Programming and Problem Solving Concepts: Programming languages - Machine level, Assembly level and high level languages, Problem solving using Algorithm and Flowchart	02	5%
2.	Fundamentals of C programming: Introduction to imperative language (ANSI C), Structure of C program, Variable Names, Data Type and Sizes, Type Conversion, Constants, Declarations, Operators, Precedence and Order of Evaluation, I/O functions	02	8%
3.	Control structures - Branching and Looping: Simple statements, Decision making statements - If statement, If-Else-If statement, Switch statement. Looping statements - for construct, while construct, do-while construct, Nesting of control structures, break and continue, goto statement.	03	20%

Sr. No.	Topics	Teaching Hrs.	Module Weightage
4.	Arrays in C: Array concepts - one-dimensional and multidimensional number and character array, declaration and initialization of arrays, string built-in functions	03	10%
5.	Functions: Basics of functions - types of functions, prototypes, calling a function, parameters, passing of parameters, return types, recursive function	03	8%
6.	Pointers: Basics of pointers, pointer and array, pointer and string, array of pointers, pointer as a function argument	02	8%
7.	Structures: Basics of structure - defining, declaring, accessing structure members, structure initialization, structures and arrays, nested structures, structure and functions	02	6%
8.	Introduction to Python: The basic elements of python, input-output, Branching Programs, Control Structures, Iteration	02	5%
9.	Structured Types, Mutability and Higher-Order Functions: Strings, Arrays, Tuples, Lists and Dictionaries, Lists and Mutability	03	9%
10.	Functions, Scoping and Abstraction: Functions and scoping, Specifications, Recursion, Global variables, Modules, Files, System Functions and Parameters, Functions as Objects	03	8%
11.	Classes and Object-Oriented Programming: Abstract Data Types and Classes, Inheritance, Encapsulation and Information Hiding	03	8%
12.	Python libraries: GPIO library, numpy, matplotlib, pandas	02	5%

Suggested Specification Table of Marks as per Bloom's Taxonomy (Theory/Practical):

% Distribution of Marks					
R Level	U Level	A Level	N Level	E Level	C Level
20	40	40	0	0	0

Legends: R: Remembrance, U: Understanding; A: Application, N: Analyze, E: Evaluate C: Create and above Levels.

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Reference Text Books:

Sr. No.	Title of book /article	Author(s)	Publisher and details like ISBN	Year of publication	Publication Edition
1.	Programming in ANSI C	Balagurusamy E	Tata McGraw-Hill Publishing Company Limited	2019	Eighth edition
2.	Let us C	Kanetkar Y. P.	BPB Publication	2016	Fifteenth edition
3.	Programming in C	B. Gottfried	Tata Mc-Graw Hill Publishers	2018	Fourth edition
4.	C Programming language	Kernighan B W and Ritchie D M	Prentice Hall	2015	Second edition
5.	Core Python Programming	R. Nageswara Rao	Dreamtech	2018	Second edition
6.	Fundamentals of Python - First Programs	Kenneth A. Lambert	CENGAGE Publication	2019	Second edition

Course Outcome:

Sr. No.	CO Statement After learning this subject, students will be able to	Marks % weightage
CO-1	Outline algorithm, flowchart and pseudocode for solving mathematical and logical problems. (R,U,A - Cognitive level)	5
CO-2	Recall the principles of computer programming using 'C' programming and Python programming language. (R,U,A - Cognitive level)	10
CO-3	Describe syntax, semantics, data types, conditional statements and control structures in 'C' and Python language. (R,U,A - Cognitive level)	35
CO-4	Exemplify the concepts of array, strings and pointers dealing with memory	25

Sr. No.	CO Statement After learning this subject, students will be able to	Marks % weightage
	management, data structures for solving computational problems. (<i>R, U, A - Cognitive level</i>)	
CO-5	Demonstrate the basic concepts of procedural programming using functions, structures, and files in writing efficient and maintainable programs. (<i>R, U, A - Cognitive level</i>)	15
CO-6	Integrate and examine object oriented programming concepts and various libraries in Python. (<i>R, U, A - Cognitive level</i>)	10

Mapping with POs:

	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
CO-1	3	2	3	2	3	1	1	3	3	2	1	3			
CO-2	2	2	2	1	2	2	1	3	3	2	1	3			
CO-3	2	2	3	1	2	2	2	3	3	2	1	3			
CO-4	2	2	3	1	2	2	2	3	3	2	1	3			
CO-5	2	2	2	1	2	2	2	3	3	2	1	3			
CO-6	3	2	3	2	3	3	3	3	3	2	1	3			
Rationale*	14	12	16	8	14	12	11	18	18	12	6	18			

***Rationale:** Explaining why it is matching this particular program outcome

LIST OF PRACTICALS:

Practical List for 'C' Programming

1. Write a program that performs as calculator (addition, multiplication, division, subtraction).
2. Write a program to find area of triangle ($a = h * b * .5$) where a = area, h = height, b = base.
3. Write a program to calculate simple interest ($i = (p * r * n) / 100$) where i = Simple interest, p = Principal amount, r = Rate of interest, n = Number of years.
4. Write a program to interchange two numbers.

Bachelor of Technology (B. Tech.)

5. Write a program to enter a distance into kilometer and convert it in to meter, feet, inches and centimeter. **Hint:** 1 Kilometer = 3280.8399 Feet, 1 kilometer = in x 0.0000254, centimeters = kilometers × 100000
6. Write a program to compute Fahrenheit from centigrade ($f = 1.8 * c + 32$)
7. Write a program to find out distance travelled by a vehicle for time (t) with equation $d = ut + at^2$ where u= velocity and a=acceleration.
8. Write a program to find that the accepted number is Negative, or Positive or Zero.
9. Write a program to read marks for a single subject of a student. If marks < 36, output the result as 'Fail' otherwise 'Pass'. (using if else)
10. Write a program to read three numbers from keyboard and find out maximum out of these three. (nested if else)
11. Write a program to check whether the entered character is capital, small letter, digit or any special character.
12. Write a program to read marks from keyboard and your program should display equivalent grade according to following table (if else ladder).

Marks	Grade
100 - 80	Distinction
79 - 60	First Class
59 - 40	Second Class
< 40	Fail

13. Write a program to prepare pay slip using following data.
Da = 10% of basic, Hra = 7.50% of basic, Ma = 300,
Pf = 12.50% of basic, Gross = basic + Da + Hra + Ma, Nt = Gross – Pf.
14. Write a program to read no 1 to 7 and print relatively day Sunday to Saturday.
15. Write a menu-driven program for scientific calculator using switch-case statement. (add, sub, mul, div, module, square, square root, power, log)
16. Write a program to print sum of first n integer numbers.
17. Write a program to check the entered character is vowel or not. (using switch...case)
18. Write a program to find out the Maximum and Minimum number from given 10 numbers using for, while loop.
19. Write a program to input an integer number and check the last digit of number is even or odd using any looping structure.
20. Write a program to print sum of individual digits of a given integer using while statement. (Use % operator)
21. Write a program to find out sum of first and last digit of a given number.

22. Write a program to check whether the given number is prime or not.
23. Write a program to print first **n** prime numbers.
24. Write a program to find factorial of a given number.
25. Write a program to generate first **n** number of Fibonacci series.
26. Write a program that accept a string and count the number of space character, tab character, new line character, and any other characters.
(Hint: use `getchar()` to accept characters. Use combination of `switch..case` and while loop).
27. Write a program to find the sum and average of different numbers. The user could be able to enter as many numbers as he wants.
28. Write a program to accept start number and end number from the user and print all the numbers in the range.
29. Write a program to calculate average and total marks of 5 students for 3 subjects (use nested *for* loops).
30. Read five persons height and weight and count the number of person having height greater than 170 and weight less than 50,
31. Write a program to evaluate the series $1^2+2^2+3^2+\dots+n^2$.
32. Write a program to find $1+1/2+1/3+1/4+\dots+1/n$.
33. Write a program to find $1+1/2!+1/3!+1/4!+\dots+1/n!$.
34. Write a program to evaluate the series $\text{sum}=1-x+x^2/2!-x^3/3!+x^4/4!-\dots-x^9/9!$.
35. Write a program to print following patterns:

1) <pre> * ** *** **** ***** </pre>	2) <pre> * * * * * * * * * * * * * * * </pre>	3) <pre> * </pre>
--	--	--

36. Write a program to print following patterns:

1) <pre> 1 12 123 1234 12345 </pre>	2) <pre> 12345 1234 123 12 1 </pre>	3) <pre> 55555 4444 333 22 1 </pre>	4) <pre> 1 22 333 4444 55555 </pre>
--	--	--	--

37. Write a program to print following patterns:

1) <pre> 1 1 2 3 1 2 3 4 5 </pre>	2) <pre> 1 1 1 1 2 1 </pre>	3) <pre> * * # * * # # * </pre>	4) <pre> 1 1 2 1 2 3 </pre>
--------------------------------------	---	---	--------------------------------

1 2 3 4 5 6 7 1 2 3 4 5 6 7 8 9	1 3 3 1 1 4 6 4 1	* # # # # * * # # # # # *	1 2 3 4 1 2 3 1 2 1
------------------------------------	----------------------	------------------------------	------------------------------

38. Write a program to print following patterns:

1) A AB ABC ABCD ABCDE	2) AAAAA BBBB CCC DD E
---------------------------------------	---------------------------------------

- 39. Write a program to find out which number is even or odd from list of 10 numbers using an array.**
- 40. Write a program to read and store the roll no and marks of 20 students using two-dimensional array.**
- 41. Write a program to sort given array in ascending order. (Use Selection sort).**
- 42. Write a program to replace a character, to delete a character in a given string.**
- 43. Write a program to reverse string.**
- 44. Write a program that defines a function to check whether a given number is prime or not.**
- 45. Write a program to find factorial of a number using recursion.**
- 46. Define a structure data type called *time_struct* containing three members' - integer hour, integer minute and integer second. Develop a program that would assign values to the individual number and display the time in the format: (16: 40: 51).**
- 47. Define a structure called 'personal' that contains person name, date of joining and salary. Using this structure, write a program to read personal information of 5 people and print the same on screen.**
- 48. Define a structure called 'cricket' that will describe the following information:
Player name, Team name, Batting average
Using cricket, declare an array 'player' with 5 elements and write a C program to read the information about all the 5 players and print team wise list containing names of players with their batting average.**
- 49. Write a program to swap two values using pointers.**
- 50. Write a program for sorting using pointer.**

Practical List for Python Programming

- 51. To display the sum of two numbers.**
- 52. Calculate area of circle.**

53. Accept a number from keyboard and test whether the given number is even or odd.
54. To display numbers from 1 to 10 using while loop.
55. To display even numbers between X and Y.
56. To display the elements of a list using for loop.
57. Draw following patterns using nested loops.

*	1	*	1
**	12	* *	2 2
***	123	* * *	3 3 3

58. Write a Python function to swap two numbers.
59. Write a Python function to find the greatest of 3 numbers.
60. Write a Python function to find the roots of a quadratic equation.
61. Write a Python function to evaluate factorial function using while loop.
62. Write a Python function to test whether a given number is prime or not.
63. Write a Python function to generate Fibonacci series till given number.
64. A python program that helps to know the effects of slicing operations on an array.
65. A python program to sort the array elements using bubble sort technique.
66. A python program to search for the position of an element in an array using index () method.
67. A python program to accept two matrices and find their product.
68. A python program to find the number of words in a string.
69. A python program to insert a sub string in a string in a particular position.
70. A python program to know how many times an element occurred in the list.
71. A python program to sort a tuple with nested tuples.
72. A python program to convert the elements of two lists into key-value pairs of a dictionary.
73. A python program to create a Bank class where deposits and withdrawals can be handled by using instance methods.
74. A python program to implement multiple inheritance using two base classes.
75. A python program to overload greater than (>) operator to make it act on class objects.
76. A python program to show method overloading to find sum of two or three numbers.
77. A python program to call super class constructor in the sub class using super ().
78. Demonstrate use of GPIO library.
79. Explore various uses of numpy.
80. Demonstrate use of matplotlib, pandas.

Major Equipment: Computer System

List of Open Source/learning website:

- <http://ps-iiith.vlabs.ac.in/>
- <https://nptel.ac.in/courses/106/104/106104128/>
 - Introduction to Programming in C.
- <https://nptel.ac.in/courses/106/106/106106145/>
 - Programming Data Structure and Algorithms using Python.
- <https://nptel.ac.in/courses/106/106/106106182/>
 - Computing using Python
- <https://www.coursera.org/>
- <https://www.udemy.com/>
- <https://www.udacity.com/>

List of Open Source Software:

1. Geany editor
2. Codeblocks editor
3. Gcc compiler
4. Anaconda
5. Pycharm